NBSIR 84-3009

BUILDING PENETRATION PROJECT

National Bureau of Standards
U.S. Department of Commerce
Boulder, Colorado 80303

September 1984

NBSIR 84-3009

BUILDING PENETRATION PROJECT

J.C. Wyss
W.J. Anson
R.D. Orr

Electromagnetic Fields Division
National Engineering Laboratory
National Bureau of Standards
U.S. Department of Commerce
Boulder, Colorado 80303

September 1984

Prepared for:
Communications Electronics Engineering Installation Agency
Fort Huachuca, Arizona 08613

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, Secretary

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

CONTENTS

ACKNOWLEDGMENTS

We would like to thank E. M. Livingston for her excellent help in performing the library searches that provided much of the data used in this project. Besides thanking J. E. Cruz and G. R. Reeve for supervising and taking the experimental data, we would also like to thank the staff at Fort Huachuca for their help in making the field measurements. Stephen Sheeran was very helpful in writing some of the printout routines, in running some of the test cases, and in graphing and drawing many of the figures. We also acknowledge the financial support from the Communications Electronics Engineering Installation Agency, Fort Huachuca, that made this study possible.

it penetrates the floor, ceiling, and each wall. The total radiant energy within a given room is then the sum of the energies arriving there via the various room-to-room routes within the building. The ratio (in dB) of this total radiant energy density to the energy density incident upon the building is the attenuation, or shielding effectiveness, assigned to that room.

1.3 Outline

Chapter two reviews the search for electrical properties of common building materials, and the mathematical expressions used to compute wall attenuation from these properties. We list the several computer data bases consulted, and note the key word groups that summoned the most useful references. Brief derivations of the computation expressions are given.

In chapter three we present the theory supporting our procedure for computing the attenuation of electromagnetic signals by buildings. It is here that we discuss the assumptions in the formulation of the problem, and limitations imposed by those assumptions.

Descriptions of the data entry programs, data files, and the computation program MASTER constitute chapter four. This material documents the programs and will be of interest mainly to the person wishing to know more about their workings. (User instructions are in the User's Guide, chapter five.)

Chapter five is a guide to the use of the programs. There are instructions on how to become a timeshare user, and how to organize data and enter it into the data files. Sample data tabulations are given; also examples of computer printouts illustrating user-computer conversations during data entry.

In chapter six are the results of building attenuation measurements made by NBS at three Army installations. Graphs present the measured data (in dB) versus frequency, and show the effect of the direction of incidence (i.e., location of the launching antenna) on the shielding effectiveness of the structures evaluated. Building floor plans show measurement locations and the placement of launching antennas.

Conclusions and bibliography are chapters seven and eight, respectively. Appendix 9.1 details procedures for making building attenuation measurements and assigning estimated uncertainty. Appendices 9.2-9.7 contains the listings for the five data entry programs, and for the computation program MASTER. Appendix 9.8 is a set of copyable forms for tabulating data to be entered into the data files.

2. SEARCH FOR AND COMPUTATION OF ELECTROMAGNETIC PROPERTIES OF BUILDING MATERIALS

2.1 Introduction

This chapter begins with a description of the literature search for data on the shielding effectiveness of building materials and of buildings themselves. The search was primarily a computer interrogation of several large data bases, although we also perused locally available journals, handbooks, reports and conference proceedings. This extensive search yielded only a few useful references, just one of which furnished us with most of our experimental data on conventional building materials.

Our computations of reflection coefficient and attenuation are based on expressions derived in most introductory texts on electromagnetic fields or electricity and magnetism. We briefly discuss these expressions and their use in obtaining the data in thirteen tables at the end of this chapter.

2.2 The Literature Search

To carry out the computer literature search, we relied on the expertise of Mrs. Victoria R. Schneller of Library Services, Environmental Research Laboratories, National Oceanic and Atmospheric Administration, Boulder, Colorado. The data bases consulted were:

- NTISearch, the computer search service of the National Technical Information Service (NTIS) of the U.S. Dept. of Commerce; accesses all technical abstracts compiled by NTIS; 1964 - present.
- INSPEC (Information Service for Physics, Electrotechnology, and Control); comprises Physics Abstracts, Electrical and Electronic Abstracts, and Computer and Control Abstracts; established by the Institute of Electrical Engineers (England); 1969 - present.
- Smithsonian Scientific Information Exchange (SSIE); 1977 - present.
- COMPENDEX, a data base of the Engineering Index Annual, a yearly publication of engineering and engineering-related abstracts; 1970 - present.
- NSA, the Standards and Specifications data base of the National Standards Association; current standards.
- SCISEARCH, the computer file of the Science Citation Index; 1974 - present.
- DTIC (Defense Technical Information Center); reports on research and development supported by the Department of Defense; 1953-present.

In the search for data on attenuation of electromagnetic waves by building materials, typical key words used were: electrical conductivity, permittivity, building material, construction material, radiofrequency, electromagnetic shielding.

The best report for experimental data on shielding effectiveness of building materials (Brennan, et al. [4]) was obtained from the NTIS data base with the combined key words: electrical properties, electrical conductivity, permittivity, building materials, electromagnetic shielding, radiofrequency, construction. A companion paper (Garrett et al. [5]) was obtained using the same key words but omitting "construction." This selection of keywords retrieved papers on both the direct measurement of electromagnetic attenuation and on measurement of electrical properties from which attenuation could be computed. The DTIC data base search gave the Brennan [4] and Garrett [5] reports in response to the combined key words: building, construction materials, dielectric properties, electrical properties, electrical conductivity. Mrs. Schneller was adept at using different combinations of key words to make an interrogation more specific.

In our search for papers on the attenuation of electromagnetic (EM) waves by buildings (as opposed to building materials), we used key word groups such as: EM field, EM radiation, EM wave absorption, attenuation, shielding, absorption, building, structure, and construction. One of the most useful papers on building attenuation (Smith [3]) had been entered in the INSPEC data base with the key words: electromagnetic compatibility, electromagnetic fields, building, shielding, radiowave propagation. As our search progressed, we found that there was very little data on radiofrequency attenuation by conventional building materials, or even on the electrical properties of such materials.

A computer search based on key words without such qualifiers as "electromagnetic" and "radiofrequency" produces references on the attenuation of nuclear as well as electromagnetic radiation. Many papers retrieved by key words specific to "radiofrequency attenuation by buildings and building materials" concerned structures hardened to electromagnetic interference by specialized construction methods and exotic materials. Often these structures were simply shielded rooms or no more than screened enclosures. Therefore, although our search was extensive, the data we use in our building attenuation computations has come from just a few reports which are identified on the data table for each material.

2.3 Computations

The three types of materials for which we computed shielding properties are dielectrics, metal sheets, and metal meshes. We discuss these computations in that order. All expressions and computed data are in SI units.

Brennan [4] measured the real and imaginary parts of the complex permittivity of common building materials as a function of frequency from 10 Hz to 1 GHz. (Because of the variable sensitivity of the measurement apparatus as the lossiness of the materials varied, measurements often could not be made at each of the intended frequencies.) From these permittivity values, we computed the power reflection coefficient and power attenuation (dB/cm) for each material in decade steps at frequencies from 10 kHz to 10

GHz. If data was not available at a desired frequency, we plotted the given data and then interpolated or extrapolated to obtain the missing point. We now briefly discuss the relations used in our computations.

The electrical characteristics of a lossy, isotropic dielectric are given by its complex permittivity (see any text, such as Johnk [6])

$$\hat{\varepsilon} = \varepsilon' - j\varepsilon'' \tag{2.1}$$

where the real part, ε', contains information on the speed and wavelength of an EM wave in the dielectric (ε' corresponds to the permittivity ε of a lossless dielectric). The imaginary term ε'' accounts for the lossy nature of the dielectric and appears in expressions for the EM wave attenuation due to those losses. Electromagnetic waves are transmitted through a dielectric (lossy or not) with the velocity $v = (\mu\varepsilon)^{-1/2}$, where μ is the permeability of the medium. (The free space values are denoted μ_0, ε_0, and $v = (\mu_0\varepsilon_0)^{-1/2} \cong 3 \times 10^8$ m/s. Air can be considered free space.) In general, ε' and ε'' are functions of the frequency of the EM wave traversing the dielectric, as observed in Brennan's data.

Given two adjoining media of permittivities $\hat{\varepsilon}_1$ and $\hat{\varepsilon}_2$, a wave in medium 1, normally incident on medium 2, has a voltage reflection coefficient ([6], p. 373)

$$\hat{\Gamma} = \frac{\hat{\eta}_2 - \hat{\eta}_1}{\hat{\eta}_2 + \hat{\eta}_1}, \qquad \hat{\eta}_i = \sqrt{\frac{\hat{\mu}_i}{\hat{\varepsilon}_i}} = \text{intrinsic wave impedance.}$$

Because $\mu_2 = \mu_0$ for most building materials in this report, and because medium 1 is air, we can write

$$\hat{\Gamma} = \frac{1 - \sqrt{\hat{\varepsilon}_r}}{1 + \sqrt{\hat{\varepsilon}_r}}, \qquad \hat{\varepsilon}_r = \hat{\varepsilon}_2/\varepsilon_0 = \text{relative permittivity,}$$

where medium 2 is the building. The power reflection coefficient is then

$$\hat{\Gamma}^2 = \left|\frac{1 - \sqrt{\hat{\varepsilon}_r}}{1 + \sqrt{\hat{\varepsilon}_r}}\right|^2 . \tag{2.2}$$

To determine if we must retain $\hat{\varepsilon}_r$ in its complex form, we write

$$\hat{\varepsilon}_r = \frac{1}{\varepsilon_0}(\varepsilon_2' - j\varepsilon_2'') = \varepsilon_r' - j\varepsilon_r''$$

$$= Ae^{-j\theta}\ ; \qquad A = [(\varepsilon_r')^2 + (\varepsilon_r'')^2]^{1/2}\ ; \qquad \theta = \tan^{-1}(\varepsilon_2''/\varepsilon_2'),$$

and so

$$(\hat{\varepsilon}_r)^{1/2} = [(\varepsilon_r')^2 + (\varepsilon_r'')^2]^{1/4}\, e^{-j\theta/2} .$$

For the materials in this report that are low-loss dielectrics ($\varepsilon'' \ll 1$), we neglect ε_r'' and approximate $\hat{\varepsilon}_r$ as the real quantity ε_r whose magnitude is ε_r'. This procedure incurs negligible error in the reflection coefficient, the worst case being 2.5% (the 30 kHz ε', ε'' values for moist clay brick) which introduces in eq (2.2) a deviation from the true reflected power of only 0.1 dB. With the above approximation for $\hat{\varepsilon}_r$, our expression for the power reflection coefficient is

$$(\Gamma)^2 = \left(\frac{1 - \sqrt{\varepsilon_r'}}{1 + \sqrt{\varepsilon_r'}}\right)^2 . \qquad (2.3)$$

As an EM wave traverses a lossy dielectric, it is attenuated; that is, the wave amplitude decreases with distance. We represent this wave as

$$E_x(z) = E_0\, e^{-\gamma z} = E_0\, e^{-(\alpha + j\beta)z} = E_0\, e^{-\alpha z}\, e^{-j\beta z}$$

where the E-field vector is in the x direction, and the wave propagates along the positive z axis. The exponential factor, $E_0\, e^{-\alpha z}$, represents the wave-amplitude attenuation with distance; α is the attenuation constant. The ratio of amplitudes at points z and $z + \ell$ is

$$\frac{E_0\, e^{-\alpha z}}{E_0\, e^{-\alpha(z + \ell)}} = e^{\alpha \ell} .$$

The attenuation of power density in the wave over the distance ℓ is the square of this ratio and is $e^{2\alpha\ell}$. In decibels, this attenuation is

$$\text{attenuation (dB)} = 10 \log e^{2\alpha\ell} = 10 \times 0.4342 \ln e^{2\alpha\ell} = 8.684\alpha\ell$$

and so

$$\text{attenuation (dB/length)} = 8.684\alpha . \qquad (2.4)$$

For a wave of angular frequency ω in a dielectric of complex permittivity $\varepsilon' - j\varepsilon''$ and permeability μ (= μ_0 for dielectrics considered here) we have (Johnk [6], p. 173)

$$\alpha = \frac{\omega\sqrt{\mu\varepsilon'}}{\sqrt{2}} \left[\sqrt{1 + (\varepsilon''/\varepsilon')^2} - 1\right]^{1/2} . \qquad (2.5)$$

Brennan has given his values of ε', ε'' as $\varepsilon'/\varepsilon_0$ (= ε_r') and $\varepsilon''/\varepsilon_0$ (= ε_r'') which he calls dielectric constant ("relative permittivity," in Johnk) and dissipation factor (Johnk's dissipation factor is $\varepsilon''/\varepsilon'$). Writing α in terms of these quantities (where $c \sim 3 \times 10^8$ m/s)

$$\alpha = \frac{\omega\sqrt{\mu_0 \varepsilon'_r \varepsilon_0}}{\sqrt{2}} \left[\sqrt{1 + \left(\frac{\varepsilon''_r}{\varepsilon'_r}\right)^2} - 1\right]^{1/2} = \frac{2\pi f\sqrt{\varepsilon'_r}}{\sqrt{2}\,c} \left[\sqrt{1 + \left(\frac{\varepsilon''_r}{\varepsilon'_r}\right)^2} - 1\right]^{1/2}$$

$$= 1.48 \times 10^{-8}\, f\, \sqrt{\varepsilon'_r} \left[\sqrt{1 + \left(\frac{\varepsilon''_r}{\varepsilon'_r}\right)^2} - 1\right]^{1/2} . \qquad (2.6)$$

In eq (2.5), the factor $\omega\sqrt{\mu\varepsilon} = 2\pi/\lambda$; thus, the dimension of α is $(\text{length})^{-1}$. In metric (S.I.) units, α has the dimension $(\text{meter})^{-1}$, and eq (2.4) will be in dB/m. For our purposes, a more reasonable dimension is dB/cm, and so our values for attenuation were computed from the expression

$$\text{attenuation (dB/cm)} = 0.08684\alpha , \qquad (2.7)$$

with α obtained from eq (2.6).

For metals and materials with significant conductivity, σ, the complex permittivity is more appropriately written

$$\hat{\varepsilon} = \varepsilon - j\,\frac{\sigma}{\omega} . \qquad (2.8)$$

Following the discussion in Johnk ([6], ch. 3), the corresponding form for α is

$$\alpha = \frac{\omega\sqrt{\mu\varepsilon}}{\sqrt{2}} \left[\sqrt{1 + \left(\frac{\sigma}{\omega\varepsilon}\right)^2} - 1\right]^{1/2} . \qquad (2.9)$$

For metals $\sigma \sim 10^7$ mhos/m. Therefore, even at 10 GHz, $\left(\frac{\sigma}{\omega\varepsilon}\right)^2 >> 1$, and α can be written

$$\alpha = \frac{\omega\sqrt{\mu\varepsilon}}{\sqrt{2}} \sqrt{\frac{\sigma}{\omega\varepsilon}} = \sqrt{\frac{\omega\mu\sigma}{2}} = \sqrt{\frac{\omega\mu_r\mu_0\sigma_r\sigma_c}{2}}$$

where μ_r is relative permeability, σ_r is conductivity relative to copper, and σ_c is the conductivity of copper. Substituting $\mu_0 = 4\pi \times 10^{-7}$ farads/meter and $\sigma_c = 5.80 \times 10^7$ mhos/meter, we have

$$\alpha = 15.13\sqrt{f\mu_r\sigma_r}$$

and so

$$\text{attenuation (dB/dm)} = 0.08684\alpha = 1.314\sqrt{f\mu_r\sigma_r} . \qquad (2.10)$$

This is the expression in Denny ([7], p. 5-6) for the attenuation of an EM wave traversing a metal sheet 1 cm thick. With eq (2.10) and values for μ_r and σ_r from Table 5-2 in Denny, we computed attenuation values for iron, copper, and aluminum sheets.

For a plane wave normally incident on a surface, the incident, reflected, and transmitted powers are related as

$$P_i = P_r + P_t ,$$

where P_t is the wave power density (w/cm^2) just across the interface and before the wave traverses any of the medium into which it has just passed. Then

$$1 = \frac{P_r}{P_i} + \frac{P_t}{P_i} = (\Gamma_r)^2 + (\Gamma_t)^2$$

and so

$$(\Gamma_r)^2 = 1 - (\Gamma_t)^2 \tag{2.11}$$

where Γ_r and Γ_t are the magnitudes of voltage reflection and transmission coefficients, and their squares are power reflection and transmission coefficients. In Denny ([7], p. 5-5), the reflection loss in dB is given as

$$R = -10 \log (\Gamma_t)^2 . \tag{2.12}$$

For iron and copper, R values are tabulated in Campi ([8], p. 28), and presented graphically for iron, copper, and aluminum in Denny (p. 5-15). Solving eq (2.12) for $(\Gamma_t)^2$, we have

$$(\Gamma_t)^2 = 10^{-R/10}$$

and we write the power reflection coefficient as

$$(\Gamma_r)^2 = 1 - 10^{-R/10} . \tag{2.13}$$

In Denny [7] and Campi [8], the smallest value of R is 57 dB from 10 kHz to 10 GHz. Therefore, we set the power reflection coefficient to unity over this frequency range for iron, copper and aluminum.

The attenuation of plane EM waves incident normally on metal wire meshes has been computed by Jakubec and Ohta [9], and we give their attenuation values in Tables 2.9 and 2.10 for galvanized steel and copper meshes. Jarva [10] has done some plane wave attenuation computations from the same equations and his values are close to those of Jakubec and Ohta. Some measured attenuations quoted by Jarva support the computed plane wave values. The theoretical expressions employed in both reports are identical, though Jarva gives their derivations.

The mesh attenuation ("insertion loss" in Jarva) computed by Jakubec and Ohta is the same quantity as R in eq (2.12). Using eq (2.13) and attenuation values in Tables 2.9 and 2.10 for R, we see that we are justified in setting the mesh reflection coefficients to unity.

The attenuation and reflection coefficients of a reinforced concrete wall are almost totally due to the reinforcing bars ("rebars") within the concrete. The low shielding effectiveness of concrete alone is seen in our computations of attenuation and reflection coefficients from Brennan's permittivity data for moist mortar (Table 2.2). ("Moist" means the mortar samples were exposed to a saturated atmosphere for one day prior to measurement. Some samples were measured dry: they were baked at 140°F for about 20 hours prior to measurement. However, the moist samples had electrical properties most similar to those of materials in field conditions.) There are many types of concrete reinforcing structures. In some cases, parallel bars without cross members are used; the bars may be horizontal or vertical. When a mesh is used, the vertical and horizontal bars may have the same spacing and the same bar diameters, or the vertical bars may be heavier and closer together. (Rebar diameters range from 0.95 cm to 6.4 cm.) Thicker walls may have two reinforcing layers: a 20-cm thick concrete wall could have a rebar layer 5 cm in from each face. Thus, knowing only that a concrete wall is reinforced, one cannot be sure of the reinforcing configuration. We use a square mesh to illustrate how rebar shielding may vary with frequency.

The equations presented in references 7, 9, and 10 for the attenuation of plane EM waves by metal screens are based on the transmission of evanescent modes through a waveguide below cutoff. However, reinforcing meshes in concrete have such large openings that cutoff occurs at much lower frequencies, well into the frequency range (10 kHz - 10 GHz) considered in this report. Therefore, we compute the power reflection coefficient for rebar meshes using an equation developed by MacFarlane ([11], p. 1527) for the voltage reflection coefficient for plane waves incident on an infinite parallel-wire grid, the plane of polarization parallel to the grid. Hill and Wait [12] obtain the same expression in their analysis of the scattering of a transient plane wave by a periodic grating. In both these treatments the scattering structure is an infinite set of equally-spaced parallel wires; there are no cross members, unlike the rebar mesh we are considering. However, Hill and Wait [13, 14] show that, for normal incidence, the reflection coefficient obtained by MacFarlane [11] and by Hill and Wait [12] is applicable to scattering from a mesh. For waves at normal incidence, the two crossed grids forming the mesh decouple and interact with the waves as separate, independent parallel-wire grids, each responding only to the E-field component along it. Thus, when the plane of polarization is aligned with one grid, the other has no interaction with the waves and drops out of the analysis. For normal incidence, the mesh field equations obtained by Hill and Wait [13, 14] yield the MacFarlane expression for the reflection coefficient of a parallel-wire grid.

The decoupling of crossed grids of parallel wires is also shown by Kontorovich [15] and Astrakhan [16], but only for the long-wavelength condition ($d \ll \lambda$, d = separation distance between wires of a grid). (Their results cannot be used for wavelengths equal to or less than d, as in our rebar computations.) The analysis contains a term proportional to the electrical resistance between the grids at the points where the wires of one grid are bonded to those of the other. The reflection coefficient for waves at normal incidence does not depend on this term so the grids are, in effect, independent elements of the mesh. For $d \ll \lambda$, MacFarlane's expression for the normal incidence reflection coefficient of a parallel-wire grid reduces to the expression for a mesh [15, 16].

For wavelengths longer than the wire spacing, the power reflection coefficient obtained from MacFarlane for 0° angle-of-incidence is

$$|\hat{\Gamma}_r|^2 = \frac{1}{1 + (\frac{2d}{\lambda})^2 \, [F(\frac{d}{\lambda}, 0^0) + \ln \frac{d}{2\pi a}]^2} \tag{2.14}$$

where d = wire spacing, a = wire radius. For $\frac{d}{\lambda} < 1$ (d = wire spacing), $F(\frac{d}{\lambda}, 0^0)$ is a real factor given graphically in references 11 and 12. For short wavelengths ($\frac{d}{\lambda} > 1$), F becomes complex ($\hat{F} = F_r + j\,F_i$), and values for F_r and F_i are also plotted [12]. The power reflection coefficient is then

$$|\hat{\Gamma}_r|^2 = \frac{1}{(1 - \frac{2d}{\lambda} F_i)^2 + (\frac{2d}{\lambda})^2 \, (F_r + \ln \frac{d}{2\pi a})^2} . \tag{2.15}$$

We have used eqs (2.14) and (2.15) to compute the plane-wave power reflection coefficient for a square rebar mesh.

Campi [8] discussed the shielding effectiveness of a rebar mesh having d = 35.6 cm and a = 2.2 cm. We used these dimensions in computing rebar power reflection coefficients from eqs (2.14) and (2.15). The definition of shielding effectiveness (SE) for any shielding material is ([7], p. 5-2)

$$SE = 10 \log \frac{P_1}{P_2} = - 10 \log \frac{P_2}{P_1} = - 10 \log (\Gamma_t)^2 = \text{attenuation (dB)}$$

where (P_1, P_2) = power density (without, with) the shield in place. Then we use eq (2.11) and compute the rebar attenuation from the expression

$$\text{attenuation (dB)} = - 10 \log (\Gamma_t)^2 = - 10 \log (1 - (\Gamma_r)^2). \tag{2.16}$$

We summarize our computation equations:

- Power reflection coefficient for a dielectric sheet: eq (2.3).
- Power attenuation within a dielectric sheet: eqs (2.6), (2.7).
- Power reflection coefficient for a metal sheet: eq (2.13).
- Power attenuation within a metal sheet: eq (2.10).
- Power reflection coefficient for a metal screen: eq (2.13), using eq (2.12).
- Power reflection coefficient for a rebar mesh: eqs (2.14), (2.15).
- Power attenuation (insertion loss) for a rebar mesh: eq (2.16).

2.4. Uncertainty

For the quantities we have computed (reflection coefficient, attenuation per unit length), we used expressions derived for plane waves at normal incidence. Into these expressions we put measured values for the real and imaginary parts of the complex permittivity. Thus, our computed quantities have uncertainties originating in the measured data we obtained in our literature search. (The reports from which this data was obtained do not give measurement uncertainties.)

However, we must include additional uncertainties in our computed data to account for other factors which influence reflection coefficients and attenuations. For example, although we have assumed normal incidence, waves may be incident at angles from 0° to 90°. A concern? Yes. Fresnel's reflection equations tell us that the reflection coefficient depends on the angle of incidence and the orientation of the plane of polarization with respect to the plane of incidence. The power in a reflected wave may also vary with surface dampness. Another factor adding to the uncertainty in our data is the dependence of wood permittivity on temperature and relative humidity [17], on the angle between the plane of incidence and the wood grain [17], and on chemicals used in treating the wood [18]. There may be other less determinate factors contributing to deviations from our computed data, factors such as manufacturing differences, age of materials, and surface weathering.

Because our computation equations were derived from plane-wave models, they contribute further to the uncertainty of our computed data. As discussed in Denny ([7], section 5.3.2), the reflection coefficient depends on the intrinsic impedance of the incident wave. Plane waves have an intrinsic impedance of about 377 Ω, while waves in the near fields of loop and dipole antennas are not planar and have lower and higher impedances, respectively. These three types of electromagnetic field are all different in their reflection loss versus frequency curves. The reflection loss of high and low impedance fields also depends on the distance of the reflecting surface from the source antenna. Thus, the reflection coefficients we have computed will differ from the true reflection coefficient when the incident wave is something other than plane.

The program MASTER computes only worst-case values for building attenuation and does not do an error analysis. The latter would be of little use considering our lack of detailed information on radiation environments and building structure and contents. However, to acknowledge the "unknowable" uncertainties introduced into our data by the various factors we have discussed, we suggest the following broad uncertainty estimates for the quantities specified:

- 1% - the essentially infinite attenuations and unity reflection coefficients of metal sheets and meshes. This small uncertainty indicates that these materials have a nearly constant effect versus frequency and changing environment.
- 10% - the attenuations of the dielectric materials (e.g., glass, brick, wood). We have assigned this higher uncertainty because these materials, nearly transparent up to microwave frequencies, are the major reason why fields so easily penetrate conventional buildings. Variations in the electrical properties of these materials will alter (though only slightly) the power density of waves passing through them into building interiors.
- 100% - the reflection coefficients of the dielectric materials. This large uncertainty should include most variations in reflection coefficient with angle of incidence, material properties, and environmental conditions.

Note that these uncertainties are in the reflected and attenuated powers and not in the decibel per centimeter values for attenuation.

When we compute the plane-wave shielding effectiveness of a layer of construction material in a wall, the reflection coefficient of the material tells us how much incident power is turned back by the layer, and the attenuation (dB/cm) tells us how much power is absorbed within the material of the layer. (By "power," we mean power density in the wave, e.g., W/cm^2.) Reflection occurs not only at the front, but also at the back surface of the layer. Depending on the reflection coefficient and the attenuation, there may be enough power in the back surface reflection that the succeeding multiple internal reflections within the layer must be considered in determining the net transmission through the layer and the net reflection from the layer. (In this regard, solid metal shields can be neglected. They have such high attenuation that little or no electromagnetic field reaches the back surface of the shield.) Common building materials are dielectrics with very low attenuation and reflection coefficients, and much of the power in the incident wave passes through such a material. Therefore, we must decide if it is sufficient to consider only the front surface reflection, or if the multiple internal reflections within a dielectric layer should be taken into account.

At any instant, the internal reflections within a dielectric sheet produce an infinite series of waves leaving the front and back surfaces of the sheet. The vector addition of the fields in these waves gives the net reflected and transmitted wave. However, to simplify the treatment and still get an estimate of the reflected and transmitted power, we assume constructive interference between all the emerging waves and so add their powers to obtain the total reflected and transmitted power. For low-loss dielectrics, we neglect attenuation within the material. This procedure gives the total reflected power

$$P_r = \frac{2(\Gamma_r)^2}{1 + (\Gamma_r)^2} P_o .$$

where P_o is the power density of the incident wave, and $(\Gamma_r)^2$ is the power reflection coefficient. The total transmitted power is

$$P_t = \frac{1 - (\Gamma_r)^2}{1 + (\Gamma_r)^2} P_o .$$

For our dielectric materials, $(\Gamma_r)^2 \ll 1$, and we can write

$$P_r \cong 2\,(\Gamma_r)^2\,[1 - (\Gamma_r)^2]\,P_o$$

and

$$P_t \cong [1 - (\Gamma_r)^2]^2\,P_o .$$

To first order in $(\Gamma_r)^2$

$$P_r \cong 2(\Gamma_r)^2 P_o$$
$$P_t \cong [1 - 2(\Gamma_r)^2] P_o. \tag{2.17}$$

For a single surface (i.e., ignoring multiple reflections caused by a second surface), the reflected and transmitted powers are obtained by inspection

$$P_r = (\Gamma_r)^2 P_o$$
$$P_t = [1 - (\Gamma_r)^2] P_o . \tag{2.18}$$

Even though the pairs of eqs (2.17) and (2.18) differ slightly in form, $(\Gamma_r)^2$ is so small that the effect of a second surface on P_r is generally negligible; and, in both cases, P_t is so slightly different from P_o that, again, we ignore the effect of a second surface. Therefore, we consider only a single reflecting/transmitting surface for each material in a building wall. Any error incurred by this assumption will be covered by the 100% uncertainty we have assigned to the reflection coefficients for dielectric construction materials.

2.5 Reflection Coefficient and Attenuation Data for Selected Building Materials

The data in the following fourteen tables has been entered into the data file MATTER and is ready for use in the building attenuation computations performed by the program MASTER. We preface the tables with these comments:

- The null material (M01) must be used as the "material" of an open doorway or an open, unscreened window. As a formality whenever material M01 is needed, the user must enter into the data file HOLES a material thickness T of 1 cm.
- Dry wall (wall board, sheet rock) is mainly plaster of Paris (material M03).
- The word "moist" (materials M02, M03, M06, M07) does not imply "soft", "fresh", or "uncured", but only that material samples measured after 24 hours in a saturated atmosphere (as opposed to samples baked dry) had electrical properties more similar to the same materials in field conditions.
- "Clay brick" (material M06) refers to the brick commonly used in homes and buildings.
- All common lumber and plywood have a very low reflection coefficient and attenuation (dB/cm). Therefore, the material data tables contain only Douglas fir and fir plywood as representative types to be used for any wood or plywood the user may encounter as building materials.
- Because the attenuation by a metal screen (materials M09, M10, M11) is actually an insertion loss given in dB instead of dB/cm, the mesh thickness is not required. However, as a formality to satisfy the computation program MASTER, the user must enter into the data file BxxxxxT a mesh thickness T of 1 cm.
- The assigned uncertainties are in the transmitted and reflected powers, and must not be applied to the attenuation in dB/cm.

Table 2.1: Null Material (Mat'l. No. M01) (Thickness T = 1 cm)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	0	0
10^5	0	0
10^6	0	0
10^7	0	0
10^8	0	0
10^9	0	0
10^{10}	0	0

Material: Fictitious; a formality, required by the program MASTER, used to represent the "material" of passageways, open doorways, and open, unscreened windows.

Table 2.2: Moist Mortar (Mat'l. No. M02)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	1.22×10^{-6}	0.23
10^5	7.54×10^{-6}	0.17
10^6	3.55×10^{-6}	0.16
10^7	7.70×10^{-5}	0.13
10^8	1.89×10^{-3}	0.10
10^9	1.12×10^{-2}	0.055
10^{10}	0.13*	0.03*

Material: Moist mortar; 6.5 gal. H_2O/94 lb. sack of cement; Portland cement-aggregate ratio: 1/3.

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.3: Plaster of Paris (Mat'l. No. M03)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	2.98×10^{-7}	0.063
10^5	4.41×10^{-7}	0.059
10^6	1.50×10^{-6}	0.084
10^7	2.58×10^{-6}	0.076
10^8	4.84×10^{-5}	0.063
10^9	7.6×10^{-4}	0.007
10^{10}	7.6×10^{-3}*	0.007*

Material: Moist plaster of Paris (main component of dry wall).

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.4: Douglas Fir (Mat'l. No. M04)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	4.28×10^{-7}	0.047
10^5	2.59×10^{-6}	0.041
10^6	4.35×10^{-6}	0.063
10^7	1.10×10^{-4}	0.050
10^8	1.98×10^{-3}	0.025
10^9	2.0×10^{-2}	0.019
10^{10}	0.22*	0.014*

Material: Douglas fir.

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.5: Fir Plywood (Mat'l. No. M05)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	1.15×10^{-6}	0.068
10^5	6.77×10^{-6}	0.048
10^6	8.31×10^{-6}	0.074
10^7	1.24×10^{-4}	0.036
10^8	2.15×10^{-3}	0.014
10^9	2.6×10^{-2}	0.013
10^{10}	0.30*	0.010*

Material: Fir plywood.

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.6: Clay Brick (Mat'l. No. M06)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	2.20×10^{-6}	0.13
10^5	1.02×10^{-5}	0.072
10^6	1.4×10^{-4}*	0.051*
10^7	2.5×10^{-3}*	0.029*
10^8	5.72×10^{-3}	0.014
10^9	5.72×10^{-3}	0.014
10^{10}	5.7×10^{-3}*	0.014*

Material: Moist clay brick.

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Interpolated or extrapolated.

Table 2.7: Cinder Block (Mat'l. No. M07)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	1.74×10^{-6}	0.17
10^5	8.30×10^{-6}	0.14
10^6	3.85×10^{-5}	0.13
10^7	2.82×10^{-4}	0.055
10^8	5.71×10^{-3}	0.013
10^9	5.71×10^{-2}	0.013
10^{10}	0.57*	0.013*

Material: Moist cinder block (Featherlite).

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.8: Glass (Mat'l. No. M08)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	1.36×10^{-8}	0.20
10^5	2.28×10^{-7}	0.18
10^6	2.88×10^{-6}	0.19
10^7	3.95×10^{-5}	0.19
10^8	4.00×10^{-4}	0.15
10^9	5.04×10^{-3}	0.082
10^{10}	7.1×10^{-2}*	0.015*

Material: Glass (type not specified). We assume "window glass" because the Brennan report concerns only building materials.

Data Source: Brennan [4].

Assigned Uncertainty: Attenuation, 10%; reflection coefficient, 100%.

*Extrapolated.

Table 2.9: Steel Mesh (Mat'l. No. M09) (Thickness T = 1 cm)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	141.0*	1.0
10^5	132.0*	1.0
10^6	114.0	1.0
10^7	94.1	1.0
10^8	44.0*	1.0
10^9	54.1	1.0
10^{10}	34.0*	1.0

Material: Galvanized steel mesh (24 × 24).

Data Source: Jakubec and Ohta [9].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

*Interpolated or extrapolated.

Table 2.10: Copper Mesh (Mat'l. No. M10) (Thickness T = 1 cm)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	106.0*	1.0
10^5	110.0*	1.0
10^6	104.2	1.0
10^7	88.0	1.0
10^8	68.0*	1.0
10^9	48.4	1.0
10^{10}	28.0*	1.0

Material: Copper mesh (20 × 20).

Data Source: Jakubec and Ohta [9].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

*Interpolated or extrapolated.

Table 2.11: Reinforcing Bar Mesh (Mat'l. No. M11) (Thickness T = 1 cm)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	1000.0*	1.0
10^5	1000.0*	1.0
10^6	1000.0*	1.0
10^7	1000.0*	1.0
10^8	12.2	0.94
10^9	0.22	0.05
10^{10}	0	0

Material: Reinforcing bar square mesh; 35.6 cm on centers; bar diameter = 4.3 cm.

Data Source: Hill and Wait [12]; MacFarlane [11].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

*To represent infinite attenuation as computed from unity power reflection coefficient.

Table 2.12: Iron Sheet (Mat'l. No. M12)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	$1.71 \times 10^{+3}$	1.0
10^5	$5.42 \times 10^{+3}$	1.0
10^6	$1.43 \times 10^{+4}$	1.0
10^7	$3.83 \times 10^{+4}$	1.0
10^8	$5.42 \times 10^{+4}$	1.0
10^9	$1.21 \times 10^{+5}$	1.0
10^{10}	$5.42 \times 10^{+4}$	1.0

Material: Iron sheet.

Data Source: Denny [7]; Campi [8].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

Table 2.13: Aluminum Sheet (Mat'l. No. M13)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	$1.03 \times 10^{+2}$	1.0
10^5	$3.24 \times 10^{+2}$	1.0
10^6	$1.03 \times 10^{+3}$	1.0
10^7	$3.24 \times 10^{+3}$	1.0
10^8	$1.03 \times 10^{+4}$	1.0
10^9	$3.24 \times 10^{+4}$	1.0
10^{10}	$1.03 \times 10^{+5}$	1.0

Material: Aluminum sheet.

Data Source: Denny [7]; Campi [8].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

Table 2.14: Copper Sheet (Mat'l. No. M14)

Frequency (Hz)	Attenuation (dB/cm)	Power reflection coefficient
10^4	$1.31 \times 10^{+2}$	1.0
10^5	$4.16 \times 10^{+2}$	1.0
10^6	$1.31 \times 10^{+3}$	1.0
10^7	$4.16 \times 10^{+3}$	1.0
10^8	$1.31 \times 10^{+4}$	1.0
10^9	$4.16 \times 10^{+4}$	1.0
10^{10}	$1.31 \times 10^{+5}$	1.0

Material: Copper sheet.

Data Source: Denny [7]; Campi [8].

Assigned Uncertainty: Attenuation, 1%; reflection coefficient, 1%.

3. CALCULATION FOR BUILDING SHIELDING EFFECTIVENESS -- THEORY

3.1 Introduction

This chapter discusses the theoretical basis for the calculation of building shielding effectiveness. After this introduction, there are four sections which consider a definition of the problem, the input data that will be used in the calculation, the actual mathematical approach used to calculate building shielding effectiveness, and window and door resonances.

3.2 Definition of Problem

The purpose of this project is to calculate how well a building shields from external electromagnetic interference. Automated on a computer, this calculation is to provide a worst case estimate of the electromagnetic power level appearing in a room of a building given the physical construction and dimensions of the building and given a known external incident field strength. The estimate must be useful over a wide frequency range (10 kHz - 10 GHz) and must be general enough so that it can be used effectively on several hundred differently constructed buildings.

3.3 Input Data Available

In any given calculation of electromagnetic shielding effectiveness, it is always desirable to have sufficient, high quality data to make the calculation yield a precision result. For example, the best data possible would be to actually measure a given building for electromagnetic shielding effectiveness using external antennas to launch a known field and calibrated antennas to measure the field inside of the building. The calculation would then be trivial -- the ratio of the launched field versus the measured internal field would be calculated directly. In fact such measurements have been performed and have yielded excellent results. Such field measurements made during this project are detailed in Chapter 6 of this report.

Direct measurements of electromagnetic shielding effectiveness, although highly effective in characterizing a building, have the disadvantage of being time consuming and expensive. A typical field measurement of one building at one site can take three or four engineers from three to five days to complete. When three to five hundred buildings are contemplated, it clearly becomes difficult to obtain actual field data in an efficient manner. Another approach is to take "less expensive" data that can be used in a model that will give an estimate of the shielding effectiveness. This estimate may not be as accurate as actual field data, but it may give sufficient accuracy to make a cost effective analysis of a building. This section describes the data that will be used in the estimation of building measurements.

For a particular calculation for a given building, the input data is restricted to the actual physical construction details of the building. The input data include: 1) building construction materials; 2) the dimensions of the building; and, 3) the construction practices involved in erecting the building. These three categories of data are to be obtained either from building drawings which were used in the original construction and which have been properly updated with modifications, or by actual physical measurements on the building itself. Although it will be less accurate than direct EMI measurements, the estimate of EM shielding effectiveness based on the physical construction of the building should be sufficiently accurate to determine the suitability of most buildings for electromagnetic shielding. In the few cases where the calculated estimate is perhaps ambiguous, actual EMI measurements could be performed. A test plan for field measurements of EMI for these marginal cases is presented in the appendix. Details and methods for gathering the actual physical data and putting it into a form usable in the present computer model will be presented in Chapter 5, "User's guide to the computer programs."

3.4 Approach

To estimate the electromagnetic shielding ability of a building, the following approach is taken. The energy entering the building from external sources is calculated for each room by considering the direct penetration through all the walls, windows and doors. The energy in each room can then go to the other rooms through internal walls and doors, as well as going back outside. As the energy continues to flow through each room, the power in each room reaches a steady state condition corresponding to a balance between the energy flowing in and out. To calculate the steady state power level, it is necessary to first consider a model that will describe the flow of the electromagnetic energy through the building. The assumptions in this model listed in figure 3.1, and graphically shown in figure 3.2, must also be considered.

The first assumption is that the input electromagnetic radiation will consist of plane waves with normal incidence on a particular external wall. The calculation will be repeated five times corresponding to the four horizontal incidence directions (i.e., North, South, East and West) and the one vertical direction (i.e., from above). The sixth direction, from below or through the ground will be ignored.

The first assumption is used to calculate the power entering a room through an external wall from the outside. If the energy density of the incident field is represented by P with units of joules/m^3, the energy, E_{IN}, entering a room in one second will be

$$E_{IN} = tPcA \tag{3.1}$$

where t is the transmission coefficient/unit length of the external wall and c is the speed of the electromagnetic radiation (the speed of light). The term A represents the area of the external wall. If the room has a door in the wall, eq (3.1) becomes

$$E_1 = t_1 P c A_1 + t_2 P c A_2 \quad (3.2)$$

where t_2 and A_2 represent the transmission coefficient and area of the door.

The second assumption is that there will be no refraction by the building of the incident plane wave, and the incoming energy will only enter those walls which are normal to the incoming radiation. This effect can also be considered a shadowing effect.

The third assumption deals with multiple layers on a wall. It states that the composite transmission coefficient of a multi-layered wall will be represented as the product of the individual transmission coefficients of the layers or

$$t_{composite} = \Pi t_i \quad (3.3)$$

where the t_i's represent the transmission coefficient of the layers. This is really a simplification that ignores internal reflections in the walls and the related interference effects.

The fourth assumption considers how the energy in a room scatters and flows to the other rooms. This assumption states that the energy in a room will be isotropically scattered in all directions. Hence, in a cubic room where the four walls, the floors and ceilings all have the same area, one sixth of the room's energy would hit each of the six surfaces equally.

To find the transmission of energy from one room to another, consider figure 3.3, a building with two rooms. The first room has an energy density N, (J/m^3). From assumption four, this energy is isotropically scattered in all directions equally. The fraction, F, of the energy scattered against the internal wall will be

$$F = \frac{A_w}{A_{1,total}} \quad , \quad (3.4)$$

where A_w is the internal wall area, and $A_{1,total}$ is the total area of the walls, floor, and ceiling in room 1. The energy that is transmitted per second from room 1 to room 2 will be

$$E_{1,2} = t\ N_1\ c\ F\ A_w \quad (3.5)$$

where the subscripts on E represent flow from room 1 to room 2. Substituting for F yields

$$E_{1,2} = t\ N_1\ c\ \frac{A_w^{\ 2}}{A_{1,t}} \quad . \quad (3.6)$$

Note that the energy flowing from room 2 to room 1 will be

$$E_{2,1} = t\ N_2\ c\ \frac{A_w^{\ 2}}{A_{2,t}} \quad . \quad (3.7)$$

which differs from eq (3.6) because the room 2 energy density, N_2, is different and the total area of room 2, $A_{2,t}$, is different. Using eqs (3.1) and (3.6) the energy transmitted between the outside to the rooms, and from room to room, can be described. We must now consider absorption and reflection for the walls, which brings us to the fifth assumption.

The fifth assumption states that wall reflections will be ignored unless the frequency of the radiation is near a resonance of the room. Outside of resonance, any radiation hitting a wall that is not transmitted will be absorbed or

$$D = 1 - t \tag{3.8}$$

where D is the absorption coefficient and t is the transmission coefficient. When the frequency is near a room resonance then reflections will be accounted for and

$$D = 1 - t - r \tag{3.9}$$

where r now represents the reflection coefficient.

The frequency range of a room resonance will be between F_{low} and F_{high} where

$$F_{low} = \frac{c}{2}\left[\frac{1}{B^2} + \frac{1}{C^2}\right]^{\frac{1}{2}}$$

and

$$F_{high} = \frac{c}{2}\left[\left(\frac{3}{A}\right)^2 + \left(\frac{3}{B}\right)^2 + \left(\frac{3}{C}\right)^2\right]^{\frac{1}{2}}$$

where A, B, and C represent the three dimensions of the room with B and C being the longest. Experience indicates that modes above 3, 3, 3 do not contribute significantly to resonance fields in a room. Therefore, these higher order modes are ignored in this model.

Note that in resonance, eq (3.9) reduces the absorption coefficient and hence reduces the effective energy loss from the room. This is equivalent to reflecting it back into the room.

The energy losses in a room can now be found by replacing t in eq (3.7) with D from either eq (3.8) or (3.9) and obtaining

$$L_{1,2} = D\, N_1\, c\, \frac{A_w^{\;2}}{A_{1,t}} \tag{3.10}$$

where L represents the energy lost from the room.

The sixth assumption states that in a calculation of the steady state energy distribution within a structure, the E or H field is computed by assuming free space impedance. To calculate the steady state energy in a room, we first write down an equation which represents the change in energy with respect to time or

$$\frac{dN_1V_1}{dt} = G - L \tag{3.11}$$

where V_1 is the volume of the room and G represents the sum of all the energy gains into the room as represented by either eqs (3.2), (3.6) or (3.7). Note that a room can gain energy from the outside (eq (3.2)) and from other rooms (eqs (3.6) or (3.7)). In eq (3.11), L represents the energy lost from a room and is taken from eq (3.6) for energy transmitted out or eq (3.10) for energy absorbed in a wall. Note that for M rooms, there will be M eq (3.11)'s.

For the steady state condition eq (3.11) will equal zero. Hence there will be M eq (3.11)'s equal to zero. Since the M eq (3.11)'s have only M unknowns (the M energy levels in each room) there will be a unique solution which gives the M energy levels in the rooms.

3.5 Input Resonances

Room resonances have already been considered by adding the reflection coefficients of a wall in eq (3.9). An additional input resonance is also considered for windows or doors in the following manner. If a window or door has a metal frame, then the transmission coefficient for the window is increased by 20 dB if the incoming frequency is near a window resonance. The frequency range for a window resonance will be defined as lying between F_{low} and F_{high} where

$$F_{low} = \frac{c}{2B}$$

$$F_{high} = \frac{c}{2} \left[\left(\frac{3}{B}\right)^2 + \left(\frac{3}{A}\right)^2 \right]^{\frac{1}{2}}$$

where A and B are the smallest and largest dimensions of the windows. Note that resonances above mode 3,3 are ignored.

1. Plane wave incident on surface(s) of structure. Use attenuation of wall material(s) to calculate transmission.
2. Ignore external refraction by building.
3. Disregard multiple internal reflections within wall(s).
4. Energy reaching inside of structure is scattered isotropically. Energy leaves structure through all available surfaces.
5. Reflections within room from walls, ceiling, etc., are only considered if frequency is near room resonance(s). Only first three resonant modes are considered. Input coupling resonances are also considered for metal door and window frames by increasing transmission by 20 dB near resonant frequency.
6. Steady state energy distribution within structure allows E or H field to be computed (free space impedance assumed).

FIGURE 3.1 ASSUMPTIONS USED IN MODEL.

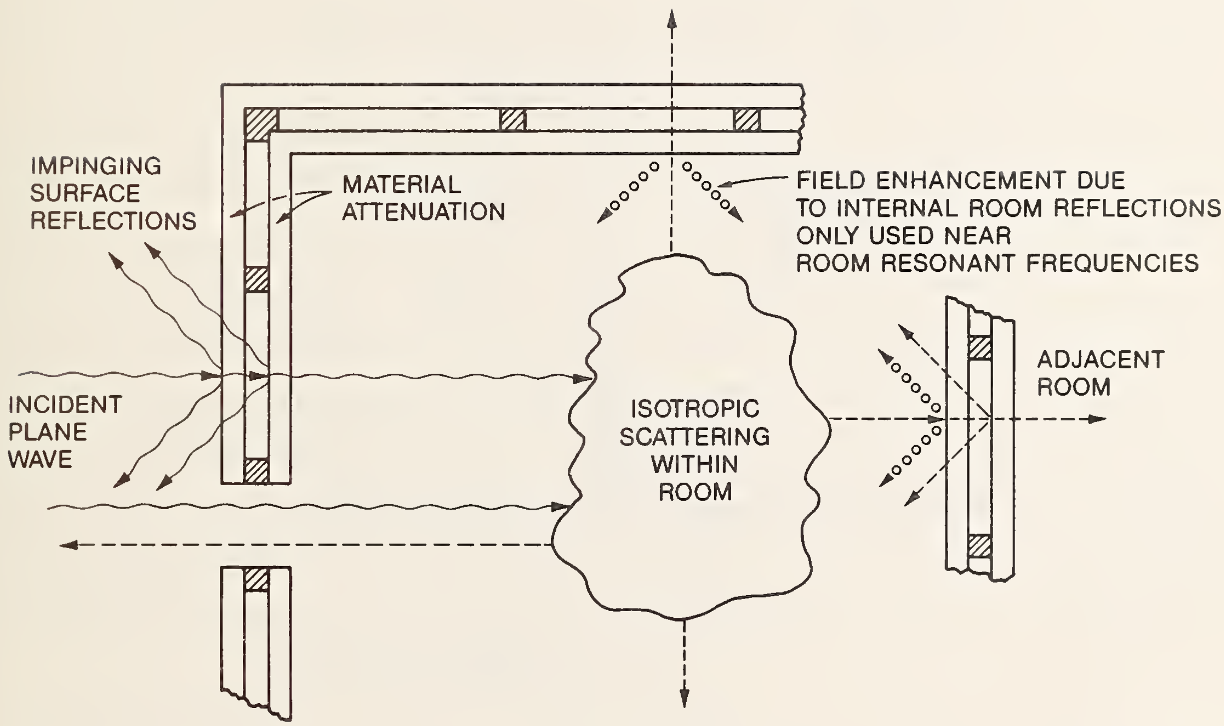


FIGURE 3.2 ENERGY FLOW ANALYSIS.

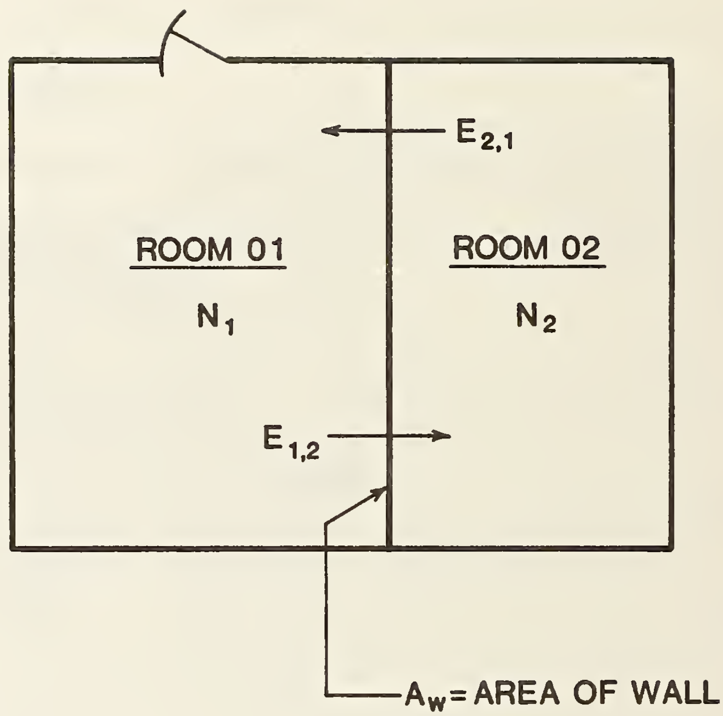


FIGURE 3.3 ENERGY FLOW BETWEEN TWO ROOMS.

4. COMPUTER PROGRAM DESCRIPTIONS

4.1 Introduction

This chapter provides a description of the computer programs used in the calculation of electromagnetic shielding effectiveness of buildings. The presentation is aimed at the programmer level. The intention is to provide information about the programming methods used in the present calculations. Information on how to use the programs is provided in Chapter 5, entitled, "User's guide to each of the programs." This chapter, besides the introduction, has four additional sections which describe the field calculation program, the data entry programs, the data structures used in the programs, and, finally, a more detailed description of the programs and their subroutines listed in alphabetical order. This last section also highlights those variables which are important to the understanding of the programs.

The project's purpose is to calculate the maximum electromagnetic field possible in each room of a building from a plane wave impinging directly upon the walls of the building given the dimensions of the building and the materials used in its construction. The program accesses permanent data files which describe the dimensions and material of each layer of each wall, door, and window of the building. These data files are created for each building using separate programs. The program also accesses a previously stored data file which contains the attenuation and reflection coefficients of different construction materials. This data is then used to calculate a transmission coefficient and an absorption coefficient for each wall considering the properties of each layer and each opening. Using these coefficients an energy flow into each room from the outside and from the other rooms is calculated. The energy flowing back out of the building is also then calculated. A steady state energy balance is then assumed for all the rooms, and from this steady state assumption, the energy level in each room is calculated.

4.2 General Description of the Field Calculation Program

A procedure file FIELD gets the main program from mass storage and runs it. The main program MASTER sets up the common blocks and then calls each of the subroutines. Five subroutines (LMATTER, LFREQ, LWALL, LTYPE, and LHOLE) load data from permanent storage into arrays which are accessed by the program when the data is needed for computation. The subroutine LTDB calculates the attenuation and area of each type opening, layer by layer, and loads it into the TDB array.

The CFACTOR subroutine calculates transmission factors of each wall and loads them into the ROOM array. These transmission factors determine the transmission between each room and its adjacent volume, which may be another room or the exterior of the building. The calculation is performed for each material layer of each wall and of each opening in the wall. Note: in this discussion "wall" also means floor and ceiling. These calculations are done sequentially and accumulated as the data for each layer of each opening is accessed in the data files.

The DFACTOR subroutine calculates the absorption factor of each wall and loads them into the DDABS array. In this calculation the absorption is taken as one minus the transmission and minus the reflection. The reflection coefficient is included if a resonance condition is met.

The subroutine ECALC is used to calculate the steady state energy in the rooms. It starts by calling the subroutine SETUP which takes the transmission coefficients from the ROOM arrays and combines them with the absorption coefficients in the DDABS array to produce the proper relationships for energy flow between the rooms and the outside. These linear relationships are put into the TMAT array. The subroutine ECALC then calculates the steady state energy balance in the rooms using the subroutine DETERM. The subroutines RESOND and RESONW calculate resonance conditions for apertures and rooms.

Several subroutines with a "P" prefix print results or file contents. For example, PHOLE, PTYPE, PWALL print the data contained in the data files and the corresponding arrays.

4.3 General Description of the Data Entry Programs

There are five data entry programs: three to store data describing buildings, one to store frequencies, and one to store properties of construction. The three which store the building data are SWALLS, STYPES, and SHOLES; the material properties are stored by SMATDB. The frequencies are stored using SFREQ. These are source code files; the user will only see the equivalent procedure files HSTORE, WSTORE, TSTORE, FSTORE and MSTORE which call and run the compiled binary equivalents of these source files.

All three building description programs use the same interactive format. Each asks the user for a building identifier which can be as many as five characters long. That building identifier is used to create the names for the permanent data files needed for each building. For example, if the building identifier were "A125" then three permanent files would be named and created: "BA125T", "BA125W", "BA125F", and "BA125H". The suffixes "T", "W", "F", and "H" refer to "Type" data, "Wall" data, "Frequency" data and "Hole" data, respectively.

The material data storage program, also interactive, stores attenuation and reflection coefficient data for different frequencies in a permanent file called MATTER. This program will only be used for adding new construction materials to the material data base.

4.4 Data Structures Used in the Programs

Arrays and Variables:

The following gives a brief description of the variables used in the common blocks.

<u>Hole Variables</u>

HMAX: maximum size of hole array. Initially set at 35.

HOLE (1-HMAX, 1-4): Array containing room and aperture ID.

HOLE (x,1): "Direction" part of room identification.

HOLE (x,2): "From room" part of room identification.

HOLE (x,3): "To room" part of room identification.

HOLE (x,4): "Aperture ID".

HTOT: A numeric variable containing the total number of lines in the "hole" data file.

HERR: A numeric variable used to indicate a file handling error and used in the WARNING and ERROR subroutines to trigger a message to the operator.

Type Variables

TMAX: Maximum size of TYPE and TDIM arrays. Initially set at 35.

TYPE (1-TMAX, 1-4): Character array containing dimensions and material of each layer of each type of door or window.

TYPE (x,1): Identification of type, e.g., "WA" meaning window "A" and "DB" meaning door B.

TYPE (x,2): Material of layer, e.g., "MØ3" meaning Material #3.

TYPE (x,3): Material of the frame, e.g., "MØ5" meaning Material #5.

TDIM (1-TMAX, 1-4): Numeric array containing dimensions of each door or window. Used in parallel with TYPE array.

TDIM (x,1): Height of opening in meters.

TDIM (x,2): Width of opening in meters.

TDIM (x,3): Thickness of layer in centimeters.

TDIM (x,4): Distance of opening above floor in meters.

TDB1 (1-TMAX): Character array containing opening identification. Used in parallel with TDB2 array.

TDB1 (x): Opening identification, e.g., "WA".

TDB2 (1-TMAX, 1-2): Numeric array containing attenuation and area of opening. Used in parallel with TDB1 array.

TDB2 (x,1): Attenuation of opening.

TDB2 (x,2): Area of opening in square meters.

TTOT: A numeric variable containing the number of lines in the "Type" data files.

TDBTOT: A numeric variable representing the total number of lines in the TDB1 and TDB2 data arrays.

TERR: A numeric variable used to indicate an error and used in the WARNING and ERROR subroutines to trigger a message to the operator.

Material Variables

MAT (1-1ØØ): Contains material identifiers such as "MØ1", "MØ2", ..."M99".

MATDESC (1-1ØØ): Contains description of each material. (7Ø characters each).

MFREQ (1-1ØØ, 1-7) real array: Contains 7 frequencies at which the property data exists.

MATTEN (1-1ØØ, 1-7): Contains 7 attenuation values for each material.

QA (1-1ØØ): Contains a quality factor for each material.

MRCOEF (1-1ØØ, 1-7): Contains 7 reflection coefficients for each material.

QR (1-1ØØ): Contains a reflection coefficient quality factor for each material.

Room Variables

DDABS (1-RMAX + 6, 1-RMAX + 6): This numeric array contains the absorption coefficients of the walls in each room.

ENERGY (1-RMAX): This numeric array contains the results of the energy balance calculations and contains the energy in each room.

POWER (1-6): This numeric array contains the power entering the building from each of the six outside directions: North, East, South, West, Top, Bottom.

RAREA (1-RMAX): Array containing surface area of each room.

RMAX: This numeric variable sets the maximum number of rooms that the program can handle. Initially set for 2Ø for simplicity of the display subroutine when printing on 8Ø column paper.

ROOM (1-RMAX + 6, 1-RMAX + 6): This numeric array is used for the transmission factors between room and room, and between room and the outside world.

TMAT (1-RMAX + 6, 1-RMAX + 6): This numeric array is the matrix that contains the linear relationships of energy flow between the rooms. It is a combination of information from the ROOM array and the DDABS array, and is created using the subroutine SETUP.

Wall Variables

WMAX: Maximum size of wall arrays. Initially set to 75.

WALL (1-WMAX, 1-4): Character array containing wall identifiers and material identification.

WALL (x,1): Direction--one of three wall identifiers.

WALL (x,2): From room--one of three wall identifiers.

Wall (x,3): To room--one of three wall identifiers.

WDIM (1-WMAX, 1-3): Numeric array containing wall dimensions.

WDIM (x,1): Height in meters.

WDIM (x,2): Width in meters.

WDIM (x,3): Layer thickness in centimeters.

WTOT: A numeric variable containing the number of lines in the Wall data files.

WERR: A numeric variable used to indicate a file handling error and used in the WARNING and ERROR subroutines to trigger a message to the operator.

Miscellaneous Variables

BLDG: This character variable contains the building identifier. It is combined with other strings to identify the various data files associated with that building.

DREFL: This variable contains the reflection coefficient for a wall. If the frequency is not near a room resonance, DREFL is set to zero.

DREFLW: This variable is used to calculate window input resonances. If the frequency is near a window resonance then DREFLW is set to 20 dB. Otherwise, DREFLW is zero.

FMAX: Maximum number of frequency values in FREQA array. Initially set at 50.

FREQ: Frequency in hertz.

FREQA (1-FMAX): A numeric array containing frequencies for calculations.

AFLAG: A number between ∅ and 1∅∅ which determines how much of the quality factor is applied to the attenuation value.

RFLAG: A number between ∅ and 1∅∅ which determines how much of the quality factor is applied to the reflection coefficient value.

Labeled Common Blocks:

This section lists variables transmitted by each labeled common block along with the subroutines using the block.

INITILC: BLDG

Common to MASTER, CFACTOR, LHOLE, LTDB, LTYPE, LWALL, PROOM, PPWR, PTMAT, DFACTOR, PDDABS, PPWR2, SPWR, RESONW, RESOND, LFREQ

INITILN: FREQ, QUALITY, AFLAG, RFLAG, FREQA (FMAX), FERR, FTOT

Common to MASTER, CFACTOR, LHOLE, LTDB, LTYPE, LWALL, PROOM, PPWR, RESOND, RESONW, PPWR2, PDDABS, PTMAT, DFACTOR, SPWR, LFREQ

HOLEC: HOLE (HMAX, 4)

Common to MASTER, CFACTOR, LHOLE, DFACTOR, PHOLE

HOLEN: HTOT, HERR

Common to MASTER, CFACTOR, LHOLE, DFACTOR, PHOLE

MATC: MAT (MMAX), MATDESC (MMAX)

Common to MASTER, ATTEN, LMATTER, RCOEF, RESONW

MAT: TMAT (RMAX, RMAX), ENERGY (RMAX), POWER (6), FTIME, SWR (RMAX, 6), IDIR

Common to MASTER, SETUP, ECALC, PPWR, PTMAT, PPWR2, SPWR, PDDABS

MATN: MATTEN (MMAX, 7), MRCOEF (MMAX, 7), QA (MMAX), QR (MMAX), MFREQ (MMAX, 7), MERR, MTOT

Common to MASTER, ATTEN, LMATTER, RCOEF, RESONW

ROOMD: DDABS (RMAX + 6, RMAX + 6), DREFL, DREFLW

Common to MASTER, CFACTOR, SETUP, DFACTOR, LDDABS, IDDABS, RESOND, RESONW

ROOMN: ROOM (RMAX + 6, RMAX + 6), NROOMS, RAREA (RMAX)

Common to MASTER, CFACTOR, LRAREA, LROOM, PROOM, SETUP, ECALC, PPWR, PTMAT, IDDABS, DFACTOR, LDDABS, PPWR2, SPWR, RESOND, RESONW, PDDABS

TYPEC: TYPE (TMAX,3), TDB1 (TMAX)

Common to MASTER, CFACTOR, LTDB, LTYPE, PTDB, PTYPE, SRCHTDB, DFACTOR, RESOND

TYPEN: TDIM (TMAX, 4), TTOT, TDB2 (TMAX, 2), TDBTOT, TERR

Common to MASTER, CFACTOR, LTDB, LTYPE, PTDB, PTYPE, SRCHTDB, DFACTOR, RESOND

WALLC: WALL (WMAX, 4)

Common to MASTER, CFACTOR, LRAREA, LWALL, PWALL, DFACTOR, RESONW

WALLN: WDIM (WMAX, 3), WTOT, WERR

Common to MASTER, CFACTOR, LRAREA, LWALL, PWALL, DFACTOR, RESONW

4.5 Description of Programs and Subroutines Used in Field Calculation (Listed Alphabetically)

Important variables (not including those passed by common blocks) are listed after each routine. Subroutines with arguments are listed with the arguments in parentheses.

*<u>FUNCTION ATTEN (ID, FREQ, AFLAG)</u>: This real function returns the material attenuation for a specified frequency and for a specified quality. It obtains the attenuation values from the MATTEN array and interpolates according to the frequency.

ID: Material identification such as "MØ1".

*<u>SUBROUTINE CFACTOR</u>: This subroutine calculates the attenuation of each wall and each opening in each wall, layer by layer, and then calculates transmission factors for each wall.

DREFLW: This value is 20 dB near a window frequency resonance.

WATTEN: Wall attenuation.

OATTEN: Opening attenuation, whether the opening is a door or window.

LATTEN: Layer attenuation.

MATTEN: Material attenuation.

MATT: Material identifier such as "MØ1".

ID: Identifier of opening.

WALL: Wall array containing Wall identification and Material identification.

*FUNCTION DETERM (ARRAY, NORDER): This real function calculates the determinant of a matrix.

ARRAY (1-RMAX + 6, 1-RMAX + 6): This real array represents the input matrix. It is destroyed during the calculation.

NORDER This real variable represents the order of the input matrix.

*SUBROUTINE DFACTOR: This subroutine calculates the attenuation of each wall and each opening in each wall, layer by layer, and then calculates the absorption factor for each wall.

DREFL: Wall reflection coefficient.

WATTEN: Wall attenuation.

OATTEN: Opening attenuation, whether the opening is a door or a window.

LATTEN: Layer attenuation.

MATTEN: Material attenuation.

ID: Identifier of opening.

WALL: Wall array containing Wall identification and Material identification.

*SUBROUTINE ECALC: This subroutine calculates the energy balance in the rooms. It calls subroutines SETUP and DETERM.

PVECTOR (1-RMAX): This real array contains the values representing the initial power levels injected into each room from the outside field.

*SUBROUTINE ERROR (IERR): This subroutine returns an error message when called with an error number as argument. It also stops the program. The error numbers and error messages are listed below:

IERR	MESSAGE
1	Materials data base is empty
2	Frequency is out of range
3	This material is not in data base
4	Denominator is zero
5	File handling error

*FUNCTION GETLEN (STRING): This integer function returns the number of characters in a character string when given the character string as an argument.

*SUBROUTINE IDDABS: This subroutine initializes the DDABS array.

*SUBROUTINE LDDABS: This subroutine loads the DDABS array.

*SUBROUTINE LFREQ: This subroutine loads the array FREQA from the permanent file "BxxxxxP", where "xxxxx" represents the building identifier.

*SUBROUTINE LHOLE: This subroutine loads the material data base from permanent storage into the HOLE array for access by the program.

*SUBROUTINE LRAREA: This subroutine calculates the surface area of each room and inserts it into the RAREA array.

*SUBROUTINE LMATTER: This subroutine loads the material data base from permanent storage into arrays for access by the program.

*SUBROUTINE LROOM (TS, TS2, FROM, TO): This subroutine loads the transmission coefficients TS and TS2 into the appropriate room location in the ROOM array.

*SUBROUTINE LTDB: This subroutine calculates the attenuation and area of each type opening, layer by layer and loads it into the TDB array (Type Data Base).

*SUBROUTINE LTYPE: This subroutine loads arrays TYPE and TDIM with data from permanent file "BxxxxxT", where "xxxxx" is the building identifier.

*SUBROUTINE LWALL: This subroutine loads arrays WALL and WDIM with data from permanent file "BxxxxxW", where "xxxxx" is the building identifier.

*PROGRAM MASTER: This program is the control section which calls each of the subroutines. The program reads in wall, window and door data; calculates transmission coefficients of each wall; stores the transmission coefficients in the ROOM matrix; and calculates the maximum field in each room on a normalized basis. It is called by the user with procedure file FIELD.

*SUBROUTINE PDDABS: This subroutine prints the ROOM matrix.

*SUBROUTINE PTDB: This subroutine prints the array TDB1 and TDB2 giving the area and attenuation of each door and window type.

*SUBROUTINE PHOLE: This subroutine prints the contents of the HOLE array giving the wall location of the doors and windows.

*SUBROUTINE PPWR: This subroutine prints the contents of the ENERGY array and represents the energy values in the rooms.

*SUBROUTINE PPWR2: A second version of PPWR which allows a more efficient format. It uses the output from the subroutine SPWR stored in the array SWR.

*SUBROUTINE PROOM: This subroutine prints the ROOM array giving the transmission factor of each wall.

*SUBROUTINE PTMAT: This subroutine prints the contents of the TMAT array. It can be used for debugging the program.

*SUBROUTINE PTYPE: This subroutine prints the contents of the arrays TYPE and TDIM giving the parameters of each door and window type.

*SUBROUTINE PWALL: This subroutine prints the contents of the WALL and WDIM arrays.

*SUBROUTINE RESOND (ID): This subroutine calculates the range of frequencies that correspond to a resonance condition for a window or door with a metal frame. If the frequency of the incoming radiation is in the range of the resonance then DREFLW is set to 20 dB. This is used in the CFACTOR subroutine to increase the transmission through a window by 20 dB if resonance occurs.

DREFLW: This is the return variable. (Used in common block ROOMD.)

FLOW: This is the lower frequency bound for resonance.

FHIGH: This is the upper frequency bound for resonance.

ID: This is the identification label for the window.

*SUBROUTINE RESONW (FROM, MATID): This subroutine calculates the frequency range that corresponds to a resonance condition in a room. If a wall has a reflection coefficient greater than 0.80 and the frequency corresponds to resonance for the room, then DREFL is set to the reflection coefficient of the wall under question. Otherwise DREFL is set to zero.

DREFL: This is the return variable. (Used in common block ROOMD.)

FLOW: This is the lower frequency bound for resonance.

FHIGH: This is the upper frequency bound for resonance.

FROM: This identifies the room being calculated.

MATID: This represents the material identification label for the wall being calculated.

*FUNCTION RCOEF (MATID, FREQ, RFLAG): This function returns the material reflection coefficient for a specified frequency and for a specified quality.

*SUBROUTINE SETUP: This subroutine loads the TMAT array.

*SUBROUTINE SPWR: This subroutine saves the energy values as they are calculated so that they can be formatted neatly when printed.

*SUBROUTINE SRCHTDB (ID, OATTEN, OAREA): This subroutine searches the TDB array given the ID label of a door or window and returns the attenuation and area of that door or window.

*FUNCTION VAL (String): This function when given a number expressed as a character string returns the number expressed as an integer.

*SUBROUTINE WARNING (IERR): This subroutine returns an error message when called with a warning number as argument. The warning number and message follows:

IERR	MESSAGE
1	HOLE data file does not exist for this bldg.
2	File handling problem on HOLE data file.
3	MATTER file does not exist for this bldg.
4	File handling problem on MATTER data file.
5	TYPE data file does not exist for this bldg.
6	File handling problem on TYPE file.
7	WALL data file does not exist for this bldg.
8	File handling problem on WALL file.
9	Height and width of room missing.
10	Length of room missing.
11	Frequency file does not exist for this building.
12	File handling problem with FREQ file.

5. USERS' GUIDE TO COMPUTER PROGRAMS FOR DATA ENTRY AND COMPUTATION

5.1 Introduction

The programs discussed in this chapter have been written for use on the Control Data Corporation Cyber 75Ø computer at the Boulder, Colorado Laboratories of the U.S. Department of Commerce. Though the programs contain checks and tests to help assure their correct use, we urge each user to read this guide carefully and to enter data (e.g., window and door types and locations) in the exact form and sequence which we specify. The checks and tests make the programs somewhat "user-friendly" but not entirely foolproof (no offense intended).

When an electromagnetic wave is normally incident on an outside wall of a building, we compute the power attenuation of the wave as it penetrates the building by a procedure comprising the data files BxxxxxF, MATTER, BxxxxxW, BxxxxxT, and BxxxxxH, and the program MASTER.

- BxxxxxF: a file containing the frequencies to be used in the calculation. The suffix "F" stands for Frequency. The "xxxxx" in the name represents the identification name of a building, e.g., B9ØØ23F would be the frequency data file for building number 9ØØ23. This convention is used for all the other data files.
- MATTER: a file containing our computed reflection coefficients and attenuation values for building materials. Users will have direct interaction with this file only if they wish to change data or enter additional data for a material already in the file, or if they wish to enter data for an additional material.
- BxxxxxW: the user enters data on the location, size, and composition of walls in the building to be evaluated. The suffix "W" stands for Walls data.
- BxxxxxT: for each door and window type, the user enters material, size, and a two-character identification. The suffix "T" stands for Types data.
- BxxxxxH: in this file, the user specifies which doors and windows are located in each wall, identifying the door and window types by their two-character codes. The suffix "H" stands for Holes data.
- MASTER: this program computes the power attenuation when an electromagnetic field, incident on an exterior wall, penetrates into any room of a building. MASTER consults the files MATTER, BxxxxxW, BxxxxxT, and BxxxxxH to obtain the material and building data necessary for the computation. It uses the file BxxxxxF to determine which frequencies to use.

All programs are written in FORTRAN V for use on the CDC 75Ø computer.

5.2 Data Preparation for Programs SWALLS, STYPES, and SHOLES

The sequence ending in an attenuation computation begins with the user drawing a plan of the building to be analyzed. This plan helps the user derive the specifications for walls, doors, and windows which are then entered into the data files BxxxxxW, BxxxxxT, and BxxxxxH.

At this point we discuss typical steps in reducing the floor plan in figure 5.1 to a set of specifications acceptable to MASTER. The procedures we illustrate with this simple example can be used in the same manner for more complex structures. The only restriction on shape is that the floor plan be rectangular or composed of adjoining rectangles. The same restriction applies to room shapes. Examine the building thoroughly to include all features (walls, doors, windows) that determine its shielding characteristics.

The building plan must be labeled as follows (see the example in fig. 5.1):

1. D1, D2, D3, D4 denote the exterior regions, or "directions", around the building. These regions must be labeled in order to specify from which direction the radiation is coming. If necessary, D5 and D6 can be the regions above and below the building, or above and below a room (e.g., a second-floor room).
2. The directions LR (left-to-right) and FB (front-to-back) specify which walls are parallel to each other. This information is useful if two parallel walls (of the same room) have high reflection coefficients, because the region between them may contain the intensified fields of standing waves produced by reflections between the walls. The program MASTER computes the highest and lowest frequencies at which these resonances may occur.
3. Label the rooms Ø1, Ø2, Ø3 . . . .
4. Determine the window and door types in the building; label these WA, WB, ..., and DA, DB, ..., respectively, at their locations on the floor plan.

To prepare data for entry into the file BxxxxxW, the user should make a data sheet to specify the size, orientation, and composition of the walls. A suggested format is shown in figure 5.2:

1. The number of each line of data is given in the "LINE#" column. When changing or entering a line of data, the user employs the line number. Note that the line numbers are shown in Figure 5.1, also.
2. The column "DIRECTION" specifies the direction to which the wall in a given line is perpendicular. This direction must be the same as that defined by the "FROM" and "TO" columns.
3. Note that lines 1, 2, 3 specify the material layers in the wall between region D4 and room Ø1, and that these layers are encountered in that sequence in going from D4 to Ø1. The material layers in a wall must be entered in the file in the correct sequence corresponding to the direction given by the "FROM" and "TO" columns.

4. The wall "HEIGHT" (meters), "LENGTH" (meters), and "THICKNESS" of material layer (centimeters) are entered in their respective columns.
5. The material identification number of each layer is entered in the "MATERIAL" column according to the material data tables in chapter 2.

We strongly advise that the user employ some means of marking the building plan as each wall is entered into the BxxxxxW data table. When the table is complete and all walls are so marked (e.g., a pencil check; a colored highlight), the user will know that none have been omitted (or perhaps entered twice). It can also be helpful to put the data table line numbers onto the drawing as they are entered into the table. In figure 5.1, these line numbers are shown at the intersections of the walls with the cross-sections (dashed lines) through the building.

To tabulate the types of doors (D) and windows (W) indicated in figure 5.1, we suggest the format in figure 5.3. From left to right, the columns specify doors and windows as follows:

1. Line numbers (the "LINE" column) are used in adding, deleting, and displaying data.
2. The floor plan in figure 5.1 has door types DA, DB, DC, and window types WA, WB.
3. A door or window has a height of H meters and a width of W meters, where H and W are the inside dimensions of a door frame and a window sash (the frame in which the glass is set).
4. "DISTANCE ABOVE FLOOR" is the distance in meters from the floor to the bottom edge of the glass or screen in a window.
5. In the next two columns, the "LAYER MATERIAL" (door or window) and its "THICKNESS" (in cm) are specified for each door and window. Notice that doors and windows may contain more than one material: window type WA has galvanized mesh screen (material MØ9) and window glass (MØ8). The user may also encounter storm windows, windows with thermopane (double-layer) glass, storm doors, and screen doors.
6. "FRAME MATERIAL"; obtain identification (e.g., MØ4) for the frame material from the material data tables.

In the specification of walls and the openings in them, all that remains now is to prepare data for BxxxxxH, the holes file which tells MASTER where the doors and windows are located. A suitable format for tabulating this data is given in figure 5.4. As before, a wall is identified by specifying 1) the two regions between which the wall is located and 2) the direction to which the wall is perpendicular (this orients the wall with respect to left-right or front-back directions). Because the wall "from D4 to Ø1" has two types of openings, it is listed twice (lines 3 and 4).

We again urge users to prepare their data for entry into the computer files by using the tabular formats we have suggested (figures 5.2, 5.3, 5.4). This procedure will reduce careless errors in transcribing building specifications from the floor plan to files BxxxxxW, BxxxxxT, and BxxxxxH.

5.3 How to Become a Remote-Site Time-Share User

Having prepared a floor plan and building data tables, the user must now enter this data into the files BxxxxxF, BxxxxxW, BxxxxxT, and BxxxxxH. Before we present details of how this is done, we describe how one becomes an off-site time-share user of the Control Data Corporation (CDC) 17Ø/75Ø computer system at the Boulder Laboratories of the U.S. Department of Commerce.

To establish a relationship with the computer, call User Services: (3Ø3) 497-5849 or (3Ø3) 497-585Ø (on FTS, 3Ø3 is a direct dial area code). When first contacting the Computer Services Division through User Services, prospective users must furnish 1) their name, organization name, and telephone number, and 2) the name and telephone number of an "authority contact" (i.e., a project leader, supervisor, etc.). User Services will then provide a user number and an initial password. When a billing account is established and the way is cleared for use of the computer, the user changes the initial password to another one which is then secure, known only to the user.

A password (4-7 alphanumeric characters) must be changed every three months, otherwise the computer will assign a new one known only to the computer. The user will then be unable to get on line without first going through User Services to submit a new password. At each log-in for about ten days prior to the expiration of a password, the computer reminds the user to enter a new one. A password is changed by the command

PASSWOR, OLDPSWD, NEWPSWD

in which the user supplies the old password (OLDPSWD) and then the new password (NEWPSWD).

To obtain an account number for computer charges, the user's purchasing department should send a purchase order to

Ms. Beverly Armstrong
NOAA/ERL, R-E52
U.S. Department of Commerce
325 Broadway
Boulder, Colorado 8Ø3Ø3.

Ms. Armstrong can be reached on (3Ø3) 497-5842 (FTS direct dial).

Dan Smith (3Ø3-497-5846) or Ron Buxton (3Ø3-497-5845) can advise which direct-dial extension to use to match the baud rate on the user's terminal.

5.4 The CDC 170/750: Data Entry and Computation

5.4.1 Log-In Procedure

Once the user is on-line, the initial conversation with the computer will be as shown in figure 5.5. In reply to the first three requests, the user enters: family name, user number, and password. If no family name is required, press the CR key (CR = carriage return). When typed, the user number does not appear on the printout. The password is typed over the blackened area. After the word "CHARGE:", the user enters

CHARGE, Z1234567,Z

(or a similar form specified by User Services) where "1234567" represents the user's account number. As seen in this printout, entering the word BYE disconnects the user from the computer.

5.4.2 Procedure File MSTORE

The file MATTER contains all the attenuation and reflection data in the tables in chapter 2. To add or change data, one uses the procedure file MSTORE (M for material), and obtains it with the BEGIN command

BEGIN, MSTORE

Figure 5.6 illustrates how the user enters building material data by means of MSTORE, which first asks if the user wants to (1) create a new data base, or (2) add to the existing data base. The user makes the choice by typing 1 or 2 after the question mark. (Note that whenever MSTORE awaits a user reply, it types a question mark as a prompt to the user.) If a data base is already present in the MATTER file and the user chooses to create a new data base, typing 1 ERASES THE EXISTING BASE. Before making this choice, the user must be certain this is indeed what is wanted. Also, one must take care not to inadvertently type the numeral 1 when 2 is intended.

The entry of material description and data is complete when the user has responded to the six prompts. (NOTE: a character string must be typed within single quotes; e.g., 'MOIST CLAY BRICK', 'M15'.) MSTORE then presents the choice of (1) adding data for another material, (2) changing data in the base, (3) displaying the data for material in the base, (4) canceling (aborting) the data set just entered, or (5) quitting the data-entry procedure. The choice (5) enters the new data into the file MATTER.

Figure 5.7 shows the user how to use MSTORE to change data already in MATTER. After the material identification MØ5 is entered, MSTORE prints all specifications for MØ5, and the user then enters the line number of the data to be changed. The procedure to change the attenuation quality percent from 1ØØ to 1Ø is then self-evident. After all changes have been made and the user has entered 99 to leave the "change" mode, the complete revised data for MØ5 is presented. Entering 5 (the "quit" option) disconnects the user from MSTORE and enters the revised data into the file MATTER. To reestablish contact with the procedure file MSTORE, the user must again enter BEGIN, MSTORE.

The entire data entry for a material can be deleted simply by using the "change" option to replace the material identification in line 1 (e.g., MØ5) with a blank. Enter two consecutive single quotes as the changed material ID; the absence of information between them is the blank.

5.4.3 Data File BxxxxxW (W for Walls)

In the file BxxxxxW, the user stores the location, dimensions, and composition of each wall in each building to be analyzed. Every building thus specified in BxxxxxW must be identified by a string of no more than five alphanumeric characters represented by xxxxx here. Building identification is the first information requested as the user begins entering or changing data in the BxxxxxW file.

Manipulating data in BxxxxxW is done through the procedure file WSTORE; that is, WSTORE is the procedure file by which the user creates and/or alters the data file BxxxxxW. Contact with BxxxxxW is therefore initiated by the command

BEGIN, WSTORE

Figure 5.8(a) illustrates the format for entering data, via WSTORE, into the file BxxxxxW. First, the user enters an identification number assigned to the building whose walls are to be documented. This identification number replaces the "xxxxx" in the file name forming a unique name. Then comes the payoff for care taken in tabulating wall data (fig. 5.2), for WSTORE now requests this data in the left-to-right order of the tabulation. Note that when data has been entered for a wall layer, entering another layer for that wall requires only "thickness of layer" and "material ID" (directional data, height, and width are the same for all layers in a given wall). If the user discontinues data entry, WSTORE presents the seven options at the top of figure 5.8(b).

Option 1: a line can be displayed as shown in figure 5.8(b). Entering the number "Ø" instead of a line number allows the user to leave the display mode.

Option 2: options 2 and 5 are similar in that they both involve adding data lines to an existing file. While 5 is only for adding a data line at the end of the file, option 2 allows the user to insert a data line between two other lines (fig. 5.8(c)). When this is done, the inserted line must be part of a wall already represented in the file (otherwise, option 5 would be used). For this reason, the user must be sure that the directions, height, and width of the inserted line match those of one (or perhaps both) of the adjacent lines (depending on whether one or both of the adjacent lines belong to the door/window of the inserted line). If the user enters the data incorrectly and the line does not match at least one of its neighbors, WSTORE will emphatically point out the error. This error message is shown at the bottom of figure 5.8(c). Note that the faulty line has not been accepted (i.e., not entered into the BxxxxxW files); the user must again choose option 2 and enter the corrected data.

Option 3: this option allows the user to delete a line of data (fig. 5.8(d)). The procedure is self-explanatory. Revised data can be entered using option 2 or 5, whichever is appropriate.

Option 4: displays all data for this building so far entered into BxxxxxW.

Option 5: adds a line of data at the end of an existing file. If the added line is another layer in the last wall in the file, the user is asked only for layer thickness and material ID. If the added line is the first layer of a new wall in the file, the format for entering the data is the same as in option 2 (fig. 5.8(c)). Neither choice within option 5 requires matching the entered data with that in an adjacent line.

Options 6 and 7: option 6 (fig. 5.8(d)) stores new data in file BxxxxxW, or replaces existing data in BxxxxxW with a revised version of that data. If the user does not wish to store the new or revised data just entered, option 7 can be used to cancel that data and leave the BxxxxxW file unchanged.

5.4.4 Data File BxxxxxT (T for Types)

In this file the user stores data from the table of door and window types and specifications (fig. 5.3). The procedure for entering this data is TSTORE, and the procedures for data entry, the cautionary comments, and the error messages are almost identical to those employed in the procedure WSTORE. Therefore, familiarity with WSTORE is sufficient warmup for using the procedure TSTORE to enter data into the file BxxxxxT. Figure 5.9 illustrates the format for data entry using TSTORE; the similarity with WSTORE is obvious.

Because doors and windows may have layers (e.g., storm doors, storm windows, screens), an insert-line-into-file option is again one of seven data-handling choices available. As in WSTORE, an inserted line of data represents an additional layer of a door or window already in the file. Therefore, the identifier (e.g., DE, WC), frame material, height, width, and distance above floor in the inserted line must match those specifications in one or both of the adjacent lines. If not, the error message and the procedure for changing the incorrect line of data are the same as in WSTORE.

The user gains access to TSTORE with the command

BEGIN, TSTORE.

Building identification is required before data entry begins.

5.4.5 Data File BxxxxxH (H for Holes)

The user stores in the file BxxxxxH the types and locations of all doors and windows in the building to be analyzed; location is given by specifying the wall containing each door and window. This data should have been previously tabulated as in figure 5.4, and is now to be entered into the data file BxxxxxH by means of the procedure HSTORE. Again, the summons is

BEGIN, HSTORE

and building identification is required to initiate data entry (fig. 5.1Ø).

Though not necessary, it may be a bookkeeping convenience to enter together the data for all doors and windows in a given wall. If option 2 ("insert line into file") is employed for this purpose, the user will be pleased to know that HSTORE does not require matching between inserted and adjacent lines. Therefore, HSTORE, while very similar to MSTORE, WSTORE, and TSTORE, is also simpler and will present no difficulties to a user familiar with the other three data entry programs.

5.4.6 Data File BxxxxxF (F for Frequency)

The user stores in the file BxxxxxF the frequencies required for the calculation using the procedure file FSTORE. The program is begun by the command,

BEGIN, FSTORE.

If no BxxxxxF file is created, the program MASTER will use a set of default frequencies in the calculation.

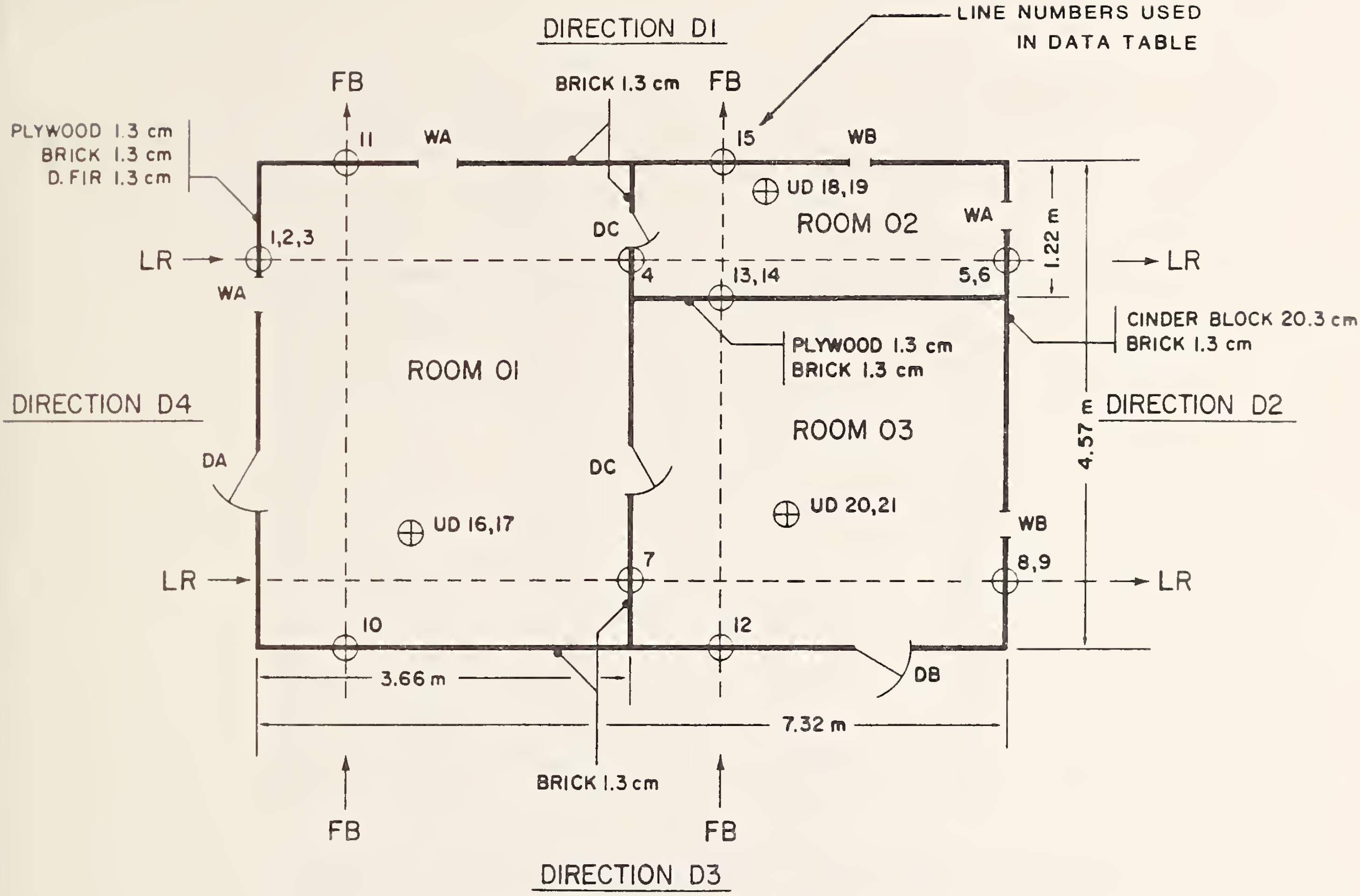
5.4.7 Computation Program MASTER

The program MASTER obtains building data from the files MATTER, BxxxxxW, BxxxxxT, and BxxxxxH, and computes the attenuation (in dB) for each room; that is, the attenuation of externally incident radiation as it penetrates into each room of the building. The user must be sure these files contain all the necessary data before consulting MASTER. When the files are ready, the user activates MASTER by entering

BEGIN, FIELD.

The procedure file FIELD summons MASTER, the data files, and the subroutines required by MASTER for its data handling operations. The only information that FIELD asks of the user is the identification code of the building to be analyzed. When the user enters this building identification, the computation begins.

For each of the five directions from which radiation may be incident on the building (e.g., north, east, south, west, above), MASTER computes all room attenuations for the frequencies given in the data file BxxxxxF. If the frequency file is missing, MASTER will use the seven default frequencies given in the material data tables. The printed output begins with a listing of the data files BxxxxxW, BxxxxxH, and BxxxxxT (fig. 5.11a); then the attenuation values are given (fig. 5.11b) by frequency, direction, and room. The data in figure 5.11 a,b are for the sample building whose wall, door, and window specifications were obtained from figure 5.1. Actual results based on field measurements are shown in chapter 7.



FLOOR PLAN OF SAMPLE BUILDING

NOTES:

WINDOW DATA

WA 1.2 m (H) x 0.91 m (W), plywood frame, 1 cm glass with 0.61 cm thick 24x24 galvanized steel mesh

WB 1.2 m (H) x 1.5 m (W), aluminum frame, 1 cm glass with 0.61 cm thick 24x24 galvanized steel mesh

DOORS

DA 2.10 m (H) x 0.91 m (W) plywood door, 5.1 cm thick with wood frame

DB 2.10 m (H) x 1.20 m (W) plywood door 5.1 cm thick with wood frame

DC 2.10 m (H) x 2.10 m (W) plywood door 1.3 cm thick with aluminum frame

WALLS

All heights are 2.44 m.

FLOOR

Cement 1.30 cm thick.

CEILING

Wood (Fir) 1.30 cm

Figure 5.1 Layout of sample building used for illustrating data input.

LINE #	DIRECTION	FROM	TO	HEIGHT	WIDTH	THICKNESS	MATERIAL
1	LR	D4	01	2.44	4.57	1.30	M05
2	LR	D4	01	2.44	4.57	1.30	M06
3	LR	D4	01	2.44	4.57	1.30	M04
4	LR	01	02	2.44	1.22	1.30	M06
5	LR	02	D2	2.44	1.22	20.30	M07
6	LR	02	D2	2.44	1.22	1.30	M06
7	LR	01	03	2.44	3.35	1.30	M06
8	LR	03	D2	2.44	3.35	20.30	M07
9	LR	03	D2	2.44	3.35	1.30	M06
10	FB	D3	01	2.00	3.66	1.30	M06
11	FB	01	D1	2.44	3.66	1.30	M06
12	FB	D3	03	2.44	3.66	1.30	M06
13	FB	03	02	2.44	3.66	1.30	M05
14	FB	03	02	2.44	3.66	1.30	M06
15	FB	02	D1	2.44	3.66	1.30	M06
16	UD	D5	01	4.57	3.66	1.30	M04
17	UD	01	D6	4.57	3.66	1.30	M02
18	UD	D5	02	1.22	3.66	1.30	M04
19	UD	02	D6	1.22	3.66	1.30	M02
20	UD	D5	03	3.35	3.66	1.30	M04
21	UD	03	D6	3.35	3.66	1.30	M02

Figure 5.2. Example of wall data tabulation for walls in figure 5.1.

LINE	ID	HEIGHT (METERS)	WIDTH (METERS)	DISTANCE ABOVE FLOOR	THICKNESS (CM)	LAYER MATERIAL	FRAME MATERIAL
1	DA	2.10	.91	0.00	5.10	M05	M05
2	DB	2.10	1.20	0.00	5.10	M05	M05
3	DC	2.10	2.10	0.00	1.30	M05	M13
4	WA	1.20	.91	.61	1.00	M08	M05
5	WA	1.20	.91	.61	.61	M09	M05
6	WB	1.20	1.50	.91	1.00	M08	M13
7	WB	1.20	1.50	.91	.61	M09	M13

Figure 5.3. Format for preparing door and window specifications for entry into file BxxxxxT.

LINE #	DIRECTION	FROM	TO	ID
1	LR	03	D2	WB
2	FB	D3	03	DB
3	LR	D4	01	WA
4	LR	D4	01	DA
5	LR	01	02	DC
6	LR	01	03	DC
7	FB	01	D1	WA
8	FB	02	D1	WB
9	LR	02	D2	WA

Figure 5.4. Format for preparing door and window location data for entry into file BxxxxxH (sample data from figure 5.1).

```
 83/09/14. 14.14.30.
N O A A / M A S C  170/750  83/06/26.   NOS 1.4 531/552.40
FAMILY:
        USER NUMBER:
                     PASSWORD
XXXXXXXX
TERMINAL:    220, TTY
RECOVER/ CHARGE: CHARGE,Z7233491,Z
/BYE

WYSS      LOG OFF    14.14.49.
WYSS      SRU      1.002 UNTS.
```

Figure 5.5. The log-in procedure.

```
/BEGIN,MSTORE.
 (1) CREATE NEW DATABASE  (2) ADD TO EXISTING DATA BASE
? 2
 (1) NEXT DATA ENTRY  (2) CHANGE  (3) DISPLAY  (4) ABORT  (5) QUIT
? 1

 MATERIAL I.D.?  (E.G.  M05 OR M12)
? 'M15'
 INDEX:15
 ENTER ONE LINE DESCRIPTION OF MATERIAL
? 'MOIST CLAY BRICK'
 ENTER 7 ATTENUATION VALUES FROM LOW TO HIGH FREQ
? .0000022,.0000102,.00014,.0025,.00572,.00572,.0057
 ENTER ATTENUATION QUALITY PERCENT
? 10.
 ENTER 7 REFLECTION COEFFS FROM LOW TO HIGH FREQ
? .13,.072,.051,.029,.014,.014,.014
 ENTER REFLECTION COEFICIENT QUALITY PERCENT
? 100.

 (1) NEXT DATA ENTRY  (2) CHANGE  (3) DISPLAY  (4) ABORT  (5) QUIT
? 5

REVERT.  MSTORE COMPLETED.
/
```

Figure 5.6. Entering building material data into the file MATTER.

```
/BEGIN,MSTORE.
 (1) CREATE NEW DATABASE  (2) ADD TO EXISTING DATA BASE
? 2
 (1) NEXT DATA ENTRY  (2) CHANGE  (3) DISPLAY  (4) ABORT  (5) QUIT
? 2

 ENTER MATERIAL I D. OF GROUP TO BE CHANGED
? 'M05'

 LINE 1      MATERIAL ID
 M04
 LINE 2      DESCRIPTION
 FIR PLYWOOD
 LINE 3:      FREQUENCIES
 10000. 100000. 1000000. 10000000. 100000000. 1000000000.
 1 E+10
 LINE 4:      ATTENUATIONS
 .00000115 .00000677 .00000831 .000124 .00215 .026 .3
 LINE 5:      ATTENUATION QUALITY PERCENT
 100.
 LINE 6:     REFLECTION COEFFICIENTS
 .068 .048 .074 .036 .014 .013 .01
 LINE 7:  REFLECTION QUALITY PERCENT
 100.

 ENTER NUMBER OF LINE TO BE CHANGED      (99 TO END CHANGES)
? 5
 ENTER NEW ATTENUATION QUALITY PERCENT
? 10.

 ENTER NUMBER OF LINE TO BE CHANGED      (99 TO END CHANGES)
? 99

 M05
 FIR PLYWOOD
 10000. 100000. 1000000. 10000000. 100000000. 1000000000.
 1.E+10
 .00000115 .00000677 .00000831 .000124 .00215 .026 .3
 10.
 .068 .048 .074 .036 .014 .013 .01
 100.

 (1) NEXT DATA ENTRY  (2) CHANGE  (3) DISPLAY  (4) ABORT  (5) QUIT
? 5

REVERT.  MSTORE COMPLETED.
/
```

Figure 5.7. Changing building material data in the file MATTER by means of the procedure file MSTORE.

```
/BEGIN,WSTORE.

 ENTER BUILDING IDENTIFICATION  (E.G. '101')
     (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? '701'

 WILL THIS BE
    (1) A MODIFICATION OF AN EXISTING FILE?
    (2) A NEW FILE?
 ENTER A NUMBER  !!!
? 2

  BEGIN ENTERING DATA

 ENTER DIRECTION (E. G. 'LR')
? 'LR'

 ENTER "FROM" (E.G. '02' OR 'D1')
? 'D4'

 ENTER "TO"  (E.G. '02' OR 'D1')
? '01'

 ENTER HEIGHT, METERS
? 2.44

 ENTER WIDTH, METERS
? 4.57

 ENTER THICKNESS OF LAYER, CENTIMETERS
? 1.30

 ENTER "MATERIAL ID" (E.G. 'M01')
? 'M05'

 DO YOU WANT TO ENTER MORE DATA?(1) YES  (2) NO
   ENTER A NUMBER !!!
? 1

 IS THIS THE FIRST LAYER OF A WALL   (1) YES (2) NO
      ENTER "0" TO ESCAPE "DATA ENTRY" MODE
  ENTER A NUMBER!!
? 2

 ENTER THICKNESS OF LAYER, CENTIMETERS
? 1.30

 ENTER "MATERIAL ID" (E.G. 'M01')
? 'M06'

 DO YOU WANT TO ENTER MORE DATA?(1) YES  (2) NO
   ENTER A NUMBER !!!
? 2

 DATA ENTRY DISCONTINUED
```

Figure 5.8(a). Procedure file WSTORE is used to enter wall data into file BxxxxxW. Data is entered in the sequence of columns in figure 5.2.

```
 CHOOSE
    (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES
    (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
    (3) DELETE LINE           (6) STORE DATA AND EXIT PROGRAM
                              (7) EXIT PROGRAM WITHOUT STORING DATA
 ENTER A NUMBER !!!

? 1

 SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED
      ( ENTER "0" TO ESCAPE DISPLAY MODE )
? 1

   LINE #   DIRECTION   FROM   TO   HEIGHT   WIDTH   THICKNESS   MATERIAL
     1         LR        D4    01    2.44     4.57      1.30        M05

 CHOOSE
    (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES
    (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
    (3) DELETE LINE           (6) STORE DATA AND EXIT PROGRAM
                              (7) EXIT PROGRAM WITHOUT STORING DATA
 ENTER A NUMBER !!!

? 6

 DOUBLE CHECK !!!
     DO YOU YOU WANT TO STORE THIS DATA AND END PROG
   NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE
   OF THE SAME NAME !!!
         ENTER A NUMBER: (1) YES   (2) NO
? 1

 DATA HAS BEEN STORED AND PROGRAM TERMINATED
REVERT.  WSTORE COMPLETED.
/
```

Figure 5.8(b). (Continued from figure 5.8(a)) How the display mode allows the user to examine a line of data.

```
/BEGIN,WSTORE.

 ENTER BUILDING IDENTIFICATION  (E.G. '101')
     (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? '701'

 WILL THIS BE
    (1) A MODIFICATON OF AN EXISTING FILE?
    (2) A NEW FILE?
 ENTER A NUMBER  !!!
? 1

 CHOOSE
    (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES
    (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
    (3) DELETE LINE           (6) STORE DATA AND EXIT PROGRAM
                              (7) EXIT PROGRAM WITHOUT STORING DATA
 ENTER A NUMBER !!!

? 2

 SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS TO BE INSERTED
     ( ENTER "0" TO ESCAPE "INSERTION" MODE )
? 2

 ENTER DIRECTION (E. G. 'LR')
? 'LR'

 ENTER "FROM" (E.G. '02' OR 'D1')
? '06'

 ENTER "TO"  (E.G. '02' OR 'D1')
? '05'

 ENTER HEIGHT, METERS
? 2.44

 ENTER WIDTH, METERS
? 2.0

 ENTER THICKNESS OF LAYER, CENTIMETERS
? 2.0

 ENTER "MATERIAL ID" (E.G. 'M01')
? 'M01'

 YOUR DATA WAS NOT ACCEPTED !!!
   YOUR DATA MUST REPRESENT A LAYER IN AN EXISTING WALL
   I.E.  THE DIRECTION, FROM, TO, HEIGHT, AND WIDTH
         PARAMETERS MUST MATCH THE WALL JUST BEFORE
         OR JUST AFTER YOUR SPECIFIED INSERTION POINT

 THE FOLLOWING DISPLAYS
 THE LINE BEFORE YOUR LINE,
 YOUR LINE, AND THE LINE AFTER

   LINE #   DIRECTION   FROM   TO   HEIGHT   WIDTH   THICKNESS   MATERIAL
      1         LR       D4    01     2.44    4.57       1.30        M05
   LINE #   DIRECTION   FROM   TO   HEIGHT   WIDTH   THICKNESS   MATERIAL
      2         LR       06    05     2.44    2.00       2.00        M01
   LINE #   DIRECTION   FROM   TO   HEIGHT   WIDTH   THICKNESS   MATERIAL
      3         LR       D4    01     2.44    4.57       1.30        M06
```

Figure 5.8(c). The WSTORE sequence for inserting data into the file BxxxxxW. Note error message when an incorrect line is inserted.

```
 CHOOSE
    (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES
    (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
    (3) DELETE LINE           (6) STORE DATA AND EXIT PROGRAM
                              (7) EXIT PROGRAM WITHOUT STORING DATA
 ENTER A NUMBER !!!

? 3

 SPECIFY THE NUMBER OF THE LINE TO BE DELETED
    (ENTER "0" TO ESCAPE DELETION MODE)
? 1

 DOUBLE CHECK !!!
      DO YOU WANT TO DELETE THE FOLLOWING LINE?:
   LINE #   DIRECTION   FROM   TO   HEIGHT   WIDTH   THICKNESS   MATERIAL
      1        LR        D4    01    2.44     4.57      1.30        M05
    ENTER (1) YES  OR   (2) NO
? 1

 LINE # 1 DELETED

 CHOOSE
    (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES
    (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
    (3) DELETE LINE           (6) STORE DATA AND EXIT PROGRAM
                              (7) EXIT PROGRAM WITHOUT STORING DATA
 ENTER A NUMBER !!!

? 6

 DOUBLE CHECK !!!
     DO YOU YOU WANT TO STORE THIS DATA AND END PROG
   NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE
   OF THE SAME NAME !!!
         ENTER A NUMBER: (1) YES   (2) NO
? 1

 DATA HAS BEEN STORED AND PROGRAM TERMINATED
REVERT.  WSTORE COMPLETED.
/
```

Figure 5.8(d). WSTORE: how to delete data (option 3), and use of option 6 to store entered data and terminate WSTORE.

```
/BEGIN,TSTORE

 ENTER BUILDING IDENTIFICATION  (E.G. '101')
    (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? 701
     701 <-ERROR IN COL.  4, RETYPE RECORD FROM THIS FIELD
? '701'

 WILL THIS BE
   (1) A MODIFICATION OF AN EXISTING FILE?
   (2) A NEW FILE?
 ENTER A NUMBER  !!!
? 2

  BEGIN ENTERING DATA

 ENTER 'ID' (E.G. 'WA' OR 'DE')
? 'WA'

 ENTER HEIGHT, METERS
? 2

 ENTER WIDTH, METERS
? .98

 ENTER DISTANCE ABOVE FLOOR, METERS
? 1

 ENTER THICKNESS OF LAYER, CENTIMETERS
? .4

 ENTER "MATERIAL ID OF LAYER" (E.G. 'M01')
? 'M05'

 ENTER "MATERIAL ID OF FRAME" (E.G. 'M01')
? 'M08'

 DO YOU WANT TO ENTER MORE DATA?(1) YES  (2) NO
   ENTER A NUMBEER !!!
? 1

 IS THIS THE FIRST LAYER OF A DOOR OR WINDOW?  (1) YES (2) NO
     ENTER "0" TO ESCAPE "DATA ENTRY" MODE
  ENTER A NUMBER!!
? 2

 ENTER THICKNESS OF LAYER, CENTIMETERS
? 1

 ENTER "MATERIAL ID OF LAYER" (E.G. 'M01')
? 'M04'

 DO YOU WANT TO ENTER MORE DATA?(1) YES  (2) NO
   ENTER A NUMBEER !!!
? 2

 DATA ENTRY DISCONTINUED

 CHOOSE
   (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES
   (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
   (3) DELETE LINE           (6) STORE DATA AND EXIT PROGRAM
                             (7) EXIT PROGRAM WITHOUT STORING DATA
 ENTER A NUMBER !!!
```

Figure 5.9. The procedure TSTORE enters data on door and window types into file BxxxxxT.

```
/BEGIN,HSTORE

 ENTER BUILDING IDENTIFICATION (E.G. '101')
     (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? '701'

 WILL THIS BE
    (1) A MODIFICATION OF AN EXISTING FILE?
    (2) A NEW FILE?
 ENTER A NUMBER  !!!
? 2

  BEGIN ENTERING DATA

 ENTER DIRECTION (E. G. 'LR')
? 'LR'

 ENTER "FROM" (E.G. '02' OR 'D1')
? 'D2'

 ENTER "TO"  (E.G. '02' OR 'D1')
? '03'

 ENTER HOLE 'ID' (E.G. 'WA' OR 'DA')
? 'WB'

 DO YOU WANT TO ENTER MORE DATA?(1) YES  (2) NO
   ENTER A NUMBER !!!
? 1

 ENTER DIRECTION (E. G. 'LR')
? 'TB'
 DIRECTION MUST BE 'LR' OR 'FB' OR 'UD'
 TRY AGAIN!!!

 ENTER DIRECTION (E. G. 'LR')
? 'FB'

 ENTER "FROM" (E.G. '02' OR 'D1')
? 'D3'

 ENTER "TO"  (E.G. '02' OR 'D1')
? '03'

 ENTER HOLE 'ID' (E.G. 'WA' OR 'DA')
? 'DB'

 DO YOU WANT TO ENTER MORE DATA?(1) YES  (2) NO
   ENTER A NUMBER !!!
? 2

 DATA ENTRY DISCONTINUED

 CHOOSE
    (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES
    (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA
    (3) DELETE LINE           (6) STORE DATA AND EXIT PROGRAM
                              (7) EXIT PROGRAM WITHOUT STORING DATA
 ENTER A NUMBER !!!

? 4

   LINE #   DIRECTION   FROM   TO   ID
      1        LR        D2    03   WB
      2        FB        D3    03   DB
```

Figure 5.10. Procedure HSTORE enters into file BxxxxxH the types and locations of doors and windows.

```
/BEGIN,FIELD
ENTER BUILDING IDENTIFICATION  (E.G. '101')
    (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? '701'
BUILDING IDENTIFICATION ENTERED AS '701'
ENTER NUMBER OF ROOMS IN BUILDING
? 3

WALL IDENTIFICATION              WALL PARAMETERS
******************** **********************************
 DIR  FROM TO          MATERIAL HEIGHT  WIDTH   THICKNESS
==================== ==================================
  LR   D4    01            M05      2.44     4.57     1.30
  LR   D4    01            M06      2.44     4.57     1.30
  LR   D4    01            M04      2.44     4.57     1.30
  LR   01    02            M06      2.44     1.22     1.30
  LR   02    D2            M07      2.44     1.22    20.30
  LR   02    D2            M06      2.44     1.22     1.30
  LR   01    03            M06      2.44     3.35     1.30
  LR   03    D2            M07      2.44     3.35    20.30
  LR   03    D2            M06      2.44     3.35     1.30
  FB   D3    01            M06      2.00     3.66     1.30
  FB   01    D1            M06      2.44     3.66     1.30
  FB   D3    03            M06      2.44     3.66     1.30
  FB   03    02            M05      2.44     3.66     1.30
  FB   03    02            M06      2.44     3.66     1.30
  FB   02    D1            M06      2.44     3.66     1.30
  UD   D5    01            M04      4.57     3.66     1.30
  UD   01    D6            M02      4.57     3.66     1.30
  UD   D5    02            M04      1.22     3.66     1.30
  UD   02    D6            M02      1.22     3.66     1.30
  UD   D5    03            M04      3.35     3.66     1.30
  UD   03    D6            M02      3.35     3.66     1.30
=======================================================

 DOOR AND WINDOW LOCATIONS
***************************
       WALL IDENTIFICATION
     ----------------------
 ID    DIRECTION  FROM  TO
==== ======================
 WB          LR     03    D2
 DB          FB     D3    03
 WA          LR     D4    01
 DA          LR     D4    01
 DC          LR     01    02
 DC          LR     01    03
 WA          FB     01    D1
 WB          FB     02    D1
 WA          LR     02    D2
===========================

               DOOR AND WINDOW PARAMETERS
**** ******************************************************
 ID   MATERIAL FRAME    HEIGHT WIDTH  LAYER      DISTANCE
               MATERIAL               THICKNESS ABOVE FLR
==== ======================================================
 DA     M05       M05     2.10      .91    5.10     0.00
 DB     M05       M05     2.10     1.20    5.10     0.00
 DC     M05       M13     2.10     2.10    1.30     0.00
 WA     M08       M05     1.20      .91    1.00      .61
 WA     M09       M05     1.20      .91     .61      .61
 WB     M08       M13     1.20     1.50    1.00      .91
 WB     M09       M13     1.20     1.50     .61      .91
=======================================================
```

Figure 5.11(a). An output of the program MASTER giving room attenuation (in dB) vs. frequency and direction of the incident radiation for building shown in figure 5.1.

ATTENUATION AT A FREQUENCY OF 1.000E+06 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-.96	-11.21	-1.34	-.41	2.23
2	1.13	-2.92	-5.32	-12.28	.89
3	-3.86	-.51	.65	-11.80	2.70

ATTENUATION AT A FREQUENCY OF 1.000E+07 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-.97	-11.22	-1.35	-.41	2.23
2	1.13	-2.93	-5.32	-12.28	.89
3	-3.87	-.52	.65	-11.80	2.70

ATTENUATION AT A FREQUENCY OF 1.000E+08 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-.37	-3.52	-.42	-2.09	2.83
2	-2.92	-6.51	-4.03	-5.75	-.45
3	-1.98	-3.33	-.78	-3.75	2.07

ATTENUATION AT A FREQUENCY OF 1.000E+09 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-.97	-12.41	-1.36	-.49	2.20
2	1.12	-4.11	-5.40	-12.40	.84
3	-3.92	-1.69	.60	-11.91	2.66

ATTENUATION AT A FREQUENCY OF 1.000E+10 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-1.02	-23.00	-1.42	-1.24	1.90
2	1.07	-14.45	-6.10	-13.56	.41
3	-4.30	-12.02	.21	-13.00	2.30

REVERT. FIELD COMPLETED.
/

Figure 5.11(b). Continuation of figure 5.11(a).

6. EXPERIMENTAL DATA

This chapter comprises reports on building attenuation measurements made by NBS at the Seneca Falls Army Depot, the Sierra Army Depot, and the Naval Training Equipment Center.

6.1 Sierra and Seneca Falls Army Depots

Equipment and personnel were assembled to make measurements of electromagnetic attenuation of three buildings at two sites selected by the Army. These sites were the Sierra Army Depot in Herlong, California, and the Seneca Army Depot in Seneca Falls, New York. The measurements were made according to the Draft Test Plan shown in Appendix 9.1, and are briefly described here. The test plan was modified for each test site by the engineers in charge to conform to any time constraints and to any constraints imposed by the physical layout of the structures. For example, the frequency range was reduced from 180 kHz - 18 GHz to 200 kHz - 10 GHz because it was decided that the additional expense involved was not justified by the present needs of the Army. Although the draft test plan recommends four physical group measurements, on the judgment of the engineers in charge, three (or in some cases, two) groups were assumed sufficient to determine the shielding effectiveness of the buildings.

The measurements were conducted over the frequency range of 200 kHz through 10 GHz. The building (No. 672) measured at the Sierra Army Depot was constructed of concrete with a metal barrier inside the wall. Two buildings were measured at the Seneca Falls Army Depot. One (No. 816) was a concrete building with one meter of dirt covering the side of the building and massive steel doors on the ends. The second building (No. 819) was constructed with cement blocks and metal doors for access.

The technique for all the measurements was as follows: a transmitting antenna was placed 30 meters from the building with the transmitted energy being directed perpendicular to the wall of the building. To get a base unattenuated signal the receiving system measured the transmitted signal outside the building at the designated frequency. The same receiving system was then used inside the building to measure the signal attenuation of the building. This signal attenuation (signal inside divided by signal outside, in dB) was measured on a predetermined distance grid within the building. Wherever possible, a $\lambda/4$ (one-quarter wavelength) offset measurement was made to detect the possibilities of building resonance effects. The receiving system used to measure these building attenuations was the isotropic antenna system developed at the National Bureau of Standards.

The attenuation measurements are summarized in tables 6.1 - 6.4 and are plotted in figures 6.1, 6.2, 6.2 and 6.4. The dimensions and physical layouts of the buildings are shown in figures 6.5, 6.6 and 6.7. The tables list the locations, the frequencies of the test, the mean values of attenuation measured, and uncertainties of the measurement (one standard deviation) shown in parentheses. Computation of the standard deviation is treated in Appendix 9.1.

6.2 Naval Training Equipment Center (NTEC)

Measurements were made of electromagnetic attenuation of three shelters connected together as a training module. The test site was located at NTEC in Orlando, Florida. The measurements were made according to the test plan shown in Appendix 9.1 and are briefly described here. The test plan was modified at the test site by the engineers in charge to conform to any constraints imposed by time or by the physical layout of the test areas.

The measurements were conducted over the frequency range of 35 MHz up to 18 GHz. As shown in figure 6.8, two ground level transmitter locations (labeled 1 and 2) were used. A third location, with the transmitter placed on the roof of an adjacent building, was used to simulate transmissions from above. The following procedure was used for all measurements. The transmitter was placed at one of the locations and was set to operate at the desired frequency and power level listed in the test plan. The receiving antenna was then placed 5 meters outside the building to measure a reference field strength. The receiving antenna was then placed inside Room 1 at the various locations shown in the test plan for the particular frequency. The attenuation (in dB) was then calculated as the measured field strength inside divided by the measured reference field strength outside. In all cases, the receiver electronics were located in Room 2 and were connected to the antennas with electrical cables.

The measured data is shown in tables 6.5 - 6.7 listed by location. The attenuation data is listed as "average", "high", and "low" values for each data point. The first label represents the average of all the data for that location, polarization, frequency, and type of field (magnetic or electric), while the other labels represent the highest and lowest value recorded.

Figures 6.9 - 6.12 show plots of the data where the circles represent electric field attenuation, while the squares represent magnetic field attenuation. The high and low values listed in the tables are shown as horizontal bars above and below each circle or square. When it would be overlapped by the size of the circle or square, the horizontal bar is not shown.

Table 6.1. EM Attenuation of Building No. 672 at Sierra Army Depot

Frequency MHz	Building Attenuation*, dB Electric Field	Magnetic Field
0.2	36 (2)	24 (3)
4	34 (5)	33 (2)
14	27 (4)	
28	33 (4)	
50	39 (6)	
140	37 (5)	
200	27 (4)	
401	22 (4)	
751	29 (3)	
998	28 (3)	
1008	27 (4)	
2000	26 (5)	
4008	39 (5)	
8007	34 (3)	

*Uncertainties representing one standard deviation are shown in parentheses.

Table 6.2. EM Attenuation of Building No. 816, End Wall at Seneca Falls, New York

Frequency MHz	Building Attenuation*, dB Electric Field	Magnetic Field
0.2	46 (6)	33 (14)
4	54 (7)	
15	57 (7)	
30	51 (8)	
50	33 (7)	
100	44 (5)	
200	44 (5)	
400	48 (13)	
750	50 (6)	
1000	45 (7)	
2000	45 (7)	
4000	42 (5)	
8000	55 (8)	

*Uncertainties representing one standard deviation are shown in parentheses.

Table 6.3. EM Attenuation of Building No. 816, Side Wall at Seneca Falls, New York

Frequency MHz	Building Attenuation*, dB Electric Field	Magnetic Field
0.2	56 (2)	50 (9)
4	49 (2)	
15	53 (4)	
30	62 (4)	
50	60 (5)	
100	56 (7)	
500	65 (12)	
1000	63 (7)	
8000	> 83†	
10,000	> 83†	

*Uncertainties representing one standard deviation are shown in parentheses.
†No measurable signal levels with available equipment.

Table 6.4. EM Attenuation of Building No. 819, End Wall at Seneca Falls, New York

Frequency MHz	Building Attenuation*, dB Electric Field	Magnetic Field
0.2	21 (3)	24 (4)
4	18 (7)	16 (5)
15	16 (2)	
30	19 (4)	
50	4 (4)	
100	10 (7)	
200	13 (7)	
500	13 (4)	
750	14 (3)	
1000	23 (4)	
4000	24 (4)	
8000	34 (18)	
10,000	35 (3)	

*Uncertainties representing one standard deviation are shown in parentheses.

Table 6.5. Attenuation measured with launch antenna at location 1, NTEC

Frequency GHz	Vertical Polarization: Electric Field Attenuation (dB)			Vertical Polarization: Magnetic Field Attenuation (dB)			Horizontal Polarization: Electric Field Attenuation (dB)		
	Average	High	Low	Average	High	Low	Average	High	Low
0.0035	-34	-43	-25	-38	-43	-34			
0.007	-23	-29	-18	-27	-33	-21			
0.014	-42	-42	-41	-45	-46	-44			
0.028	-43	-49	-37	-43	-46	-40			
0.054	-26	-26	-26						
0.088	-17	-18	-15						
0.14	-15	-16	-13				-16	-18	-14
0.20	-19	-21	-16				-13	-15	-11
0.40	-23	-25	-18				-10	-15	- 5
0.75	-20	-23	-19				-31	-31	-30
1.0	-10	-16	- 0				-10	-13	- 7
2.0	- 7	- 8	- 5				- 7	- 8	- 4
8.0	-37	-38	-35				-23	-26	-20
12.0									
18.0	-11	-11	-11				-13	-15	-12

Table 6.6. Attenuation measured with launch antenna at location 2, NTEC

Frequency GHz	Vertical Polarization Electric Field Attenuation (dB)			Vertical Polarization Magnetic Field Attenuation (dB)			Horizontal Polarization Electric Field Attenuation (dB)		
	Average	High	Low	Average	High	Low	Average	High	Low
0.0035	-22.5	-25	-20	-21	-21	-20			
0.007	-30	-40	-21	-20	-20	-19			
0.014	-27	-28	-26	-25	-27	-23			
0.028	-28	-33	-22	-26	-34	-18			
0.054	-32	-39	-25						
0.088	-13	-17	-10						
0.14	- 6	- 8	- 1				-13	-14	-11
0.20	- 7	-10	- 5				-17	-19	-14
0.40	-12	-14	-10				- 7	-11	- 4
0.75	-13	-16	-11				-18	-20	-16
1.0	-22	-24	-21				-20	-26	-17
2.0	-31	-34	-27				-22	-26	-19
8.0	-17	-17	-16				-19	-20	-18
12.0	-13	-24	- 1				-15	-21	- 9
18.0									

Table 6.7. Attenuation measured with launch antenna at location 3 (roof), NTEC

Frequency GHz	Vertical Polarization Electric Field Attenuation (dB)			Horizontal Polarization Electric Field Attenuation (dB)		
	Average	High	Low	Average	High	Low
8.0	-20	-22	-18	-20	-22	-17

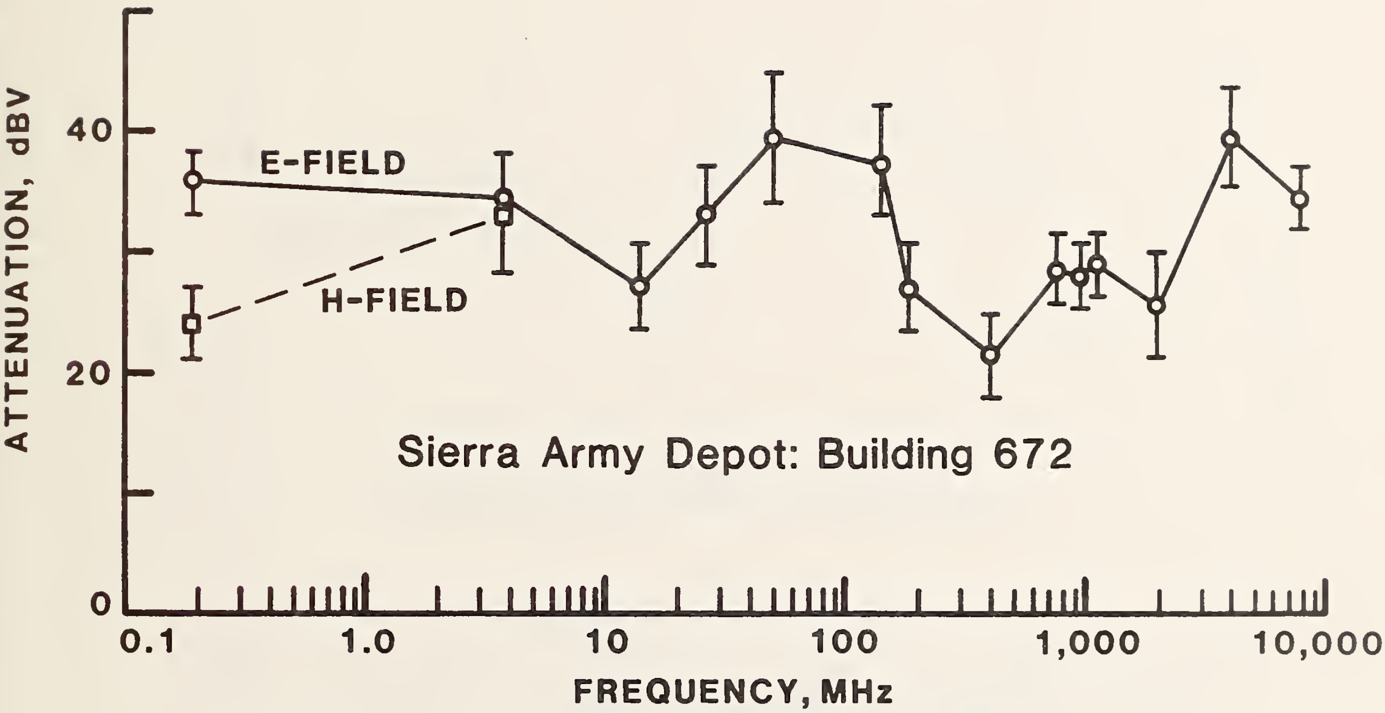


Figure 6.1. Electromagnetic attenuation versus frequency of Building No. 672 at Sierra Army Depot. Electric field attenuation (open circles) was measured from Ø.2 - 1Ø,ØØØ MHz while magnetic field (boxes) was measured at Ø.2 and 4 MHz. Error bars represent one standard deviation. See figure 6.5 for locations of transmitter and receivers.

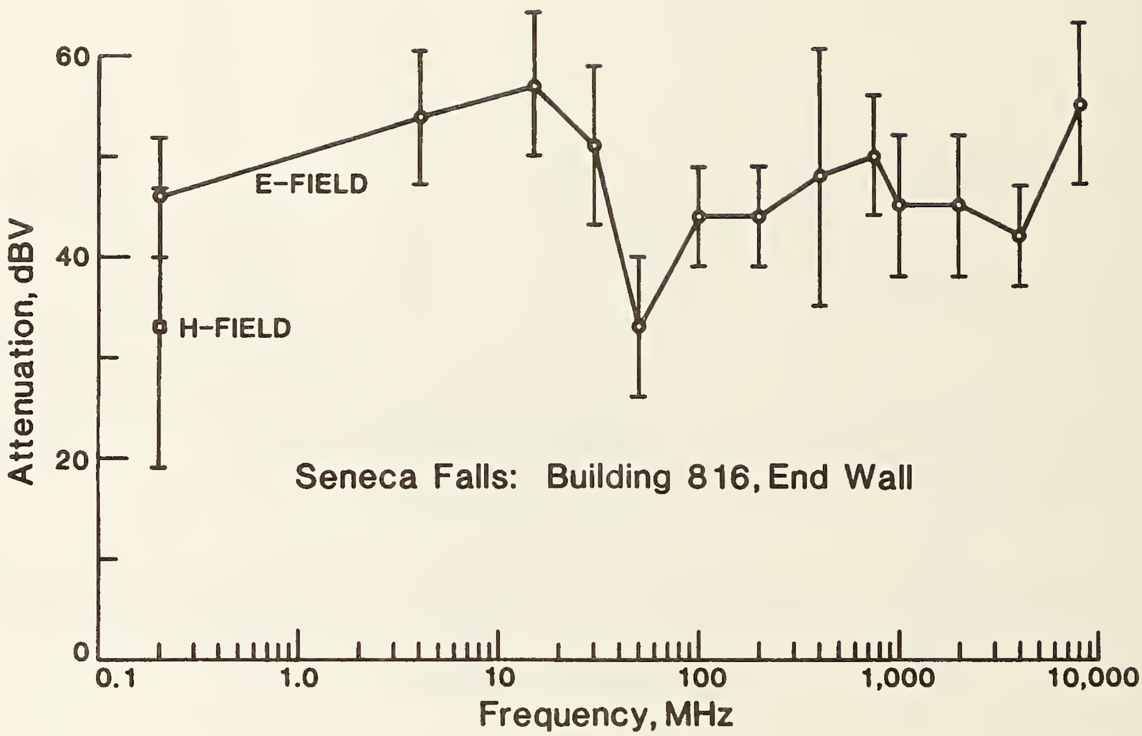


Figure 6.2. Electromagnetic attenuation versus frequency for Building No. 816 at Seneca Falls. The data are plotted as figure 6.1. For this scan, the transmitter was located at the "End Wall" as shown in figure 6.6.

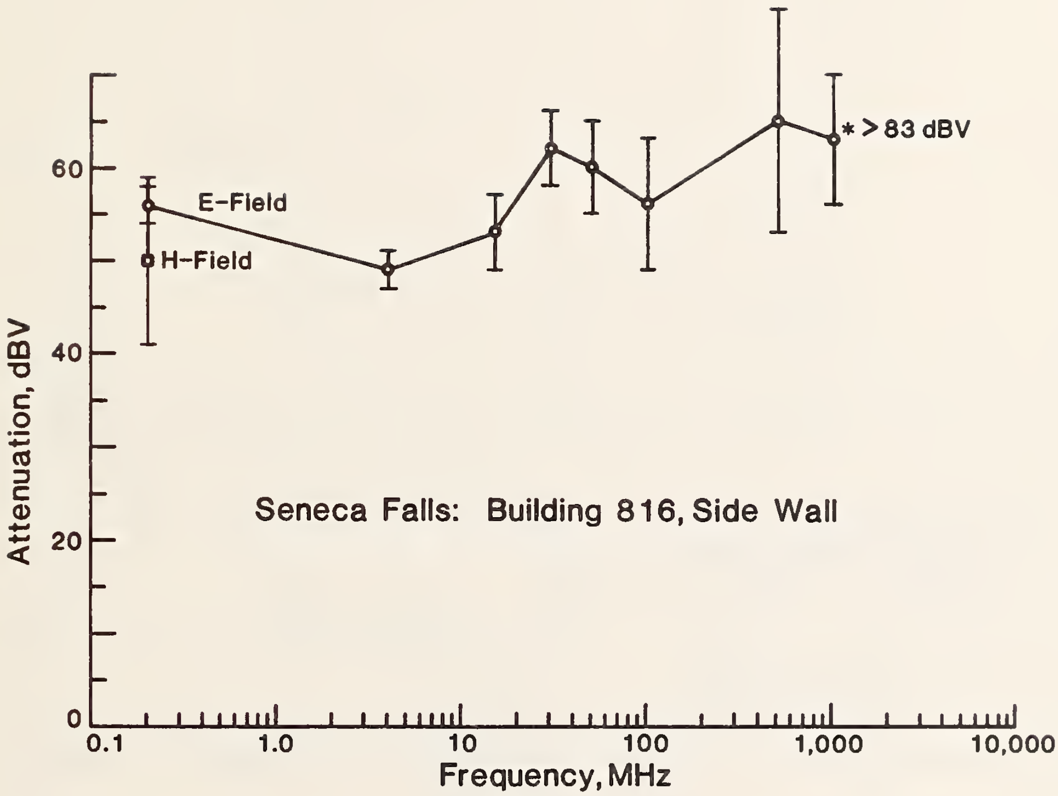


Figure 6.3. Electromagnetic attenuation versus frequency for Building No. 816 at Seneca Falls. The data are plotted as in figure 6.1. For this scan, the transmitter was located at the "Side Wall" as shown in figure 6.6. Magnetic field attenuation was measured at Ø.2 MHz, only.

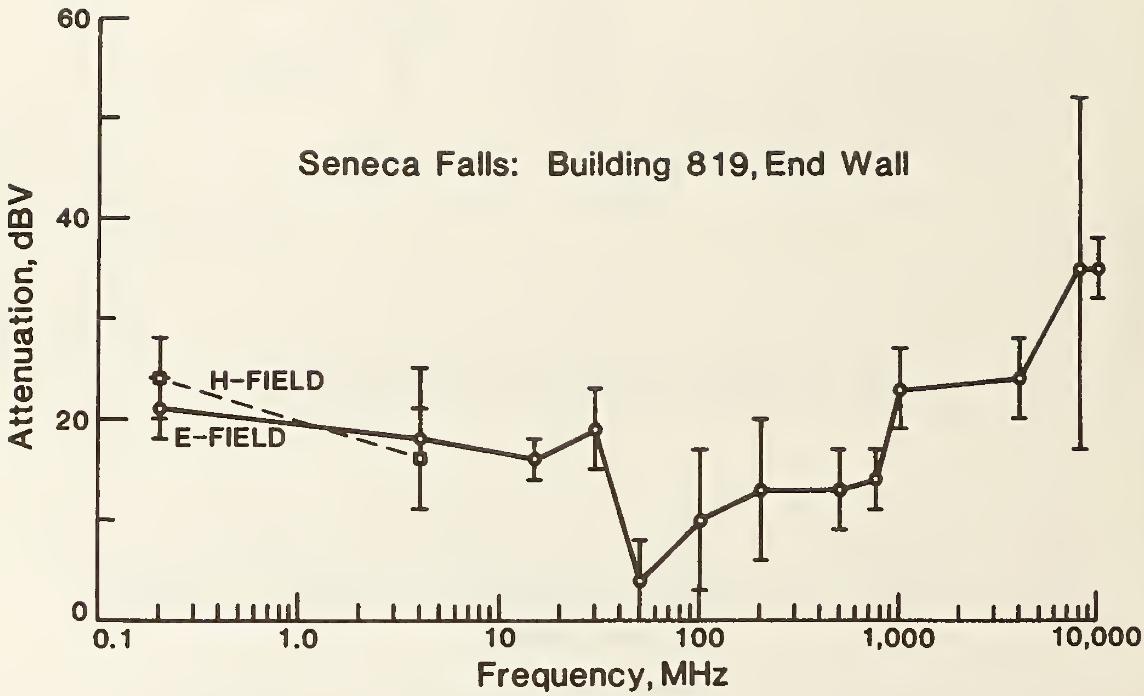


Figure 6.4. Electromagnetic attenuation versus frequency for Building No. 819 at Seneca Falls. The data are plotted the same as in figure 6.1. The transmitter and receiver locations are shown in figure 6.7. Magnetic field attenuation was measured at Ø.2 and 4 MHz.

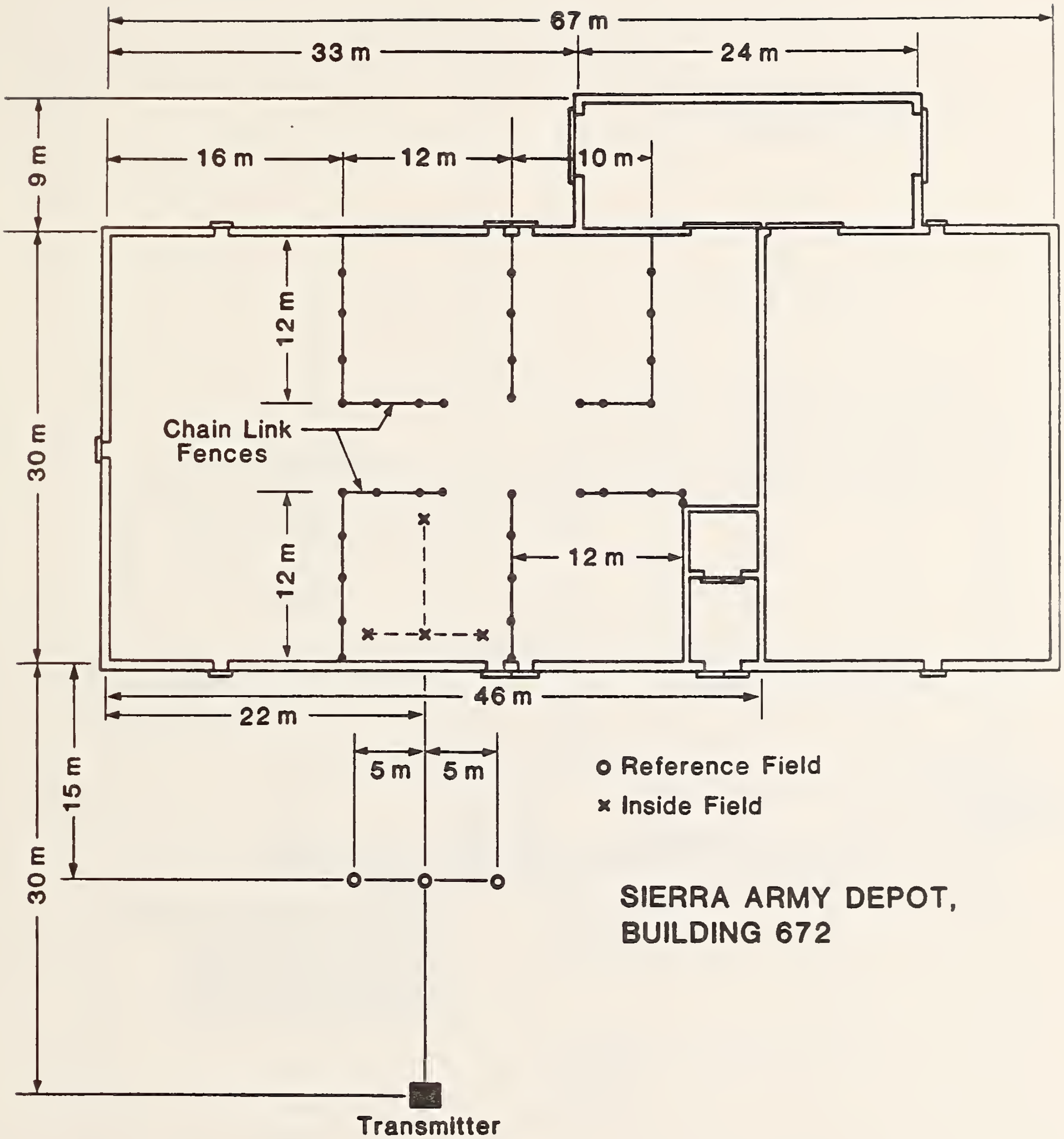


Figure 6.5. Physical layout of Building No. 672 at Sierra Army Depot. The transmitter is located 3Ø meters from the building and is shown as a square box. Outside reference fields were measured 15 meters from the building at the locations marked with open circles. One row and one column of inside measurements were made at the locations marked by two dashed lines with x's. Measurements are made at one meter intervals along the dashed line.

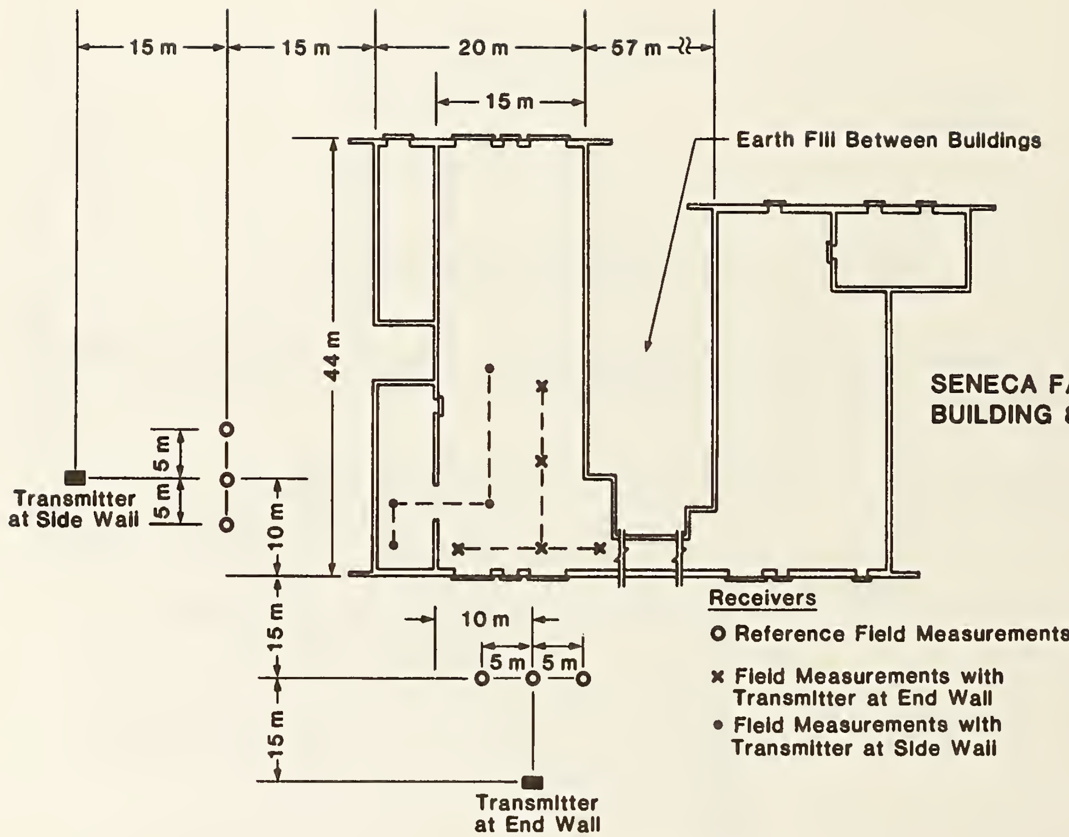


Figure 6.6. Physical layout of Building No. 816 at Seneca Falls. For this site, two sets of data were taken with the transmitter located at the two boxes marked on the drawing. Reference, outside measurements were taken at the locations marked with open circles while inside measurements were taken along the dashed line. For the end wall transmitter location, inside measurements were taken at the locations marked with x's; while the side wall transmitter measurements are marked with closed circles. Measurements are made at one meter intervals along the dashed lines.

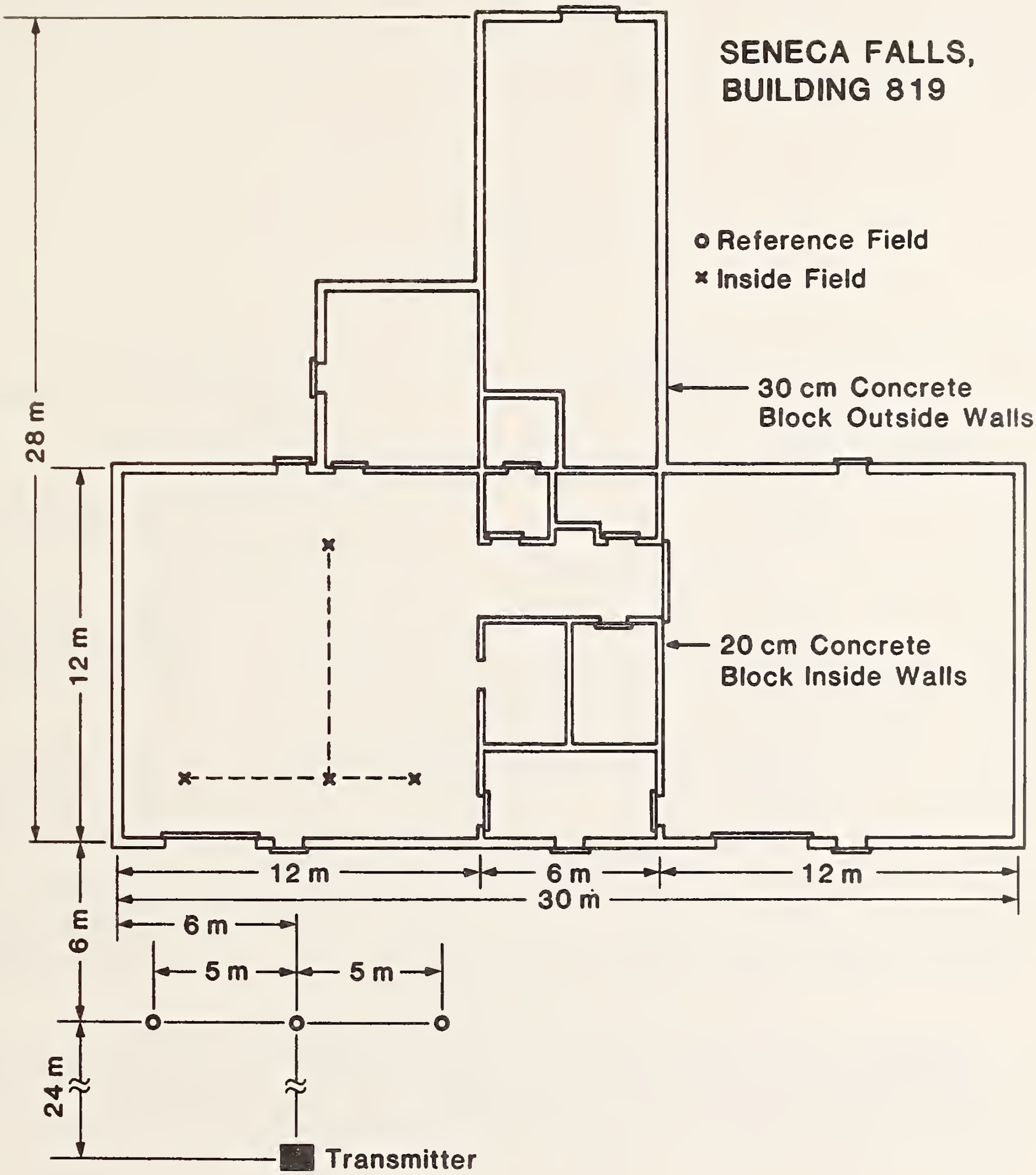


Figure 6.7. Physical layout of Building No. 819 at Seneca Falls. The transmitter is located at the solid box marked on the drawing, while the outside reference points are shown as open circles. The inside measurement points are located along the dashed lines with the x's marked. Measurements are made at one meter intervals along the dashed lines.

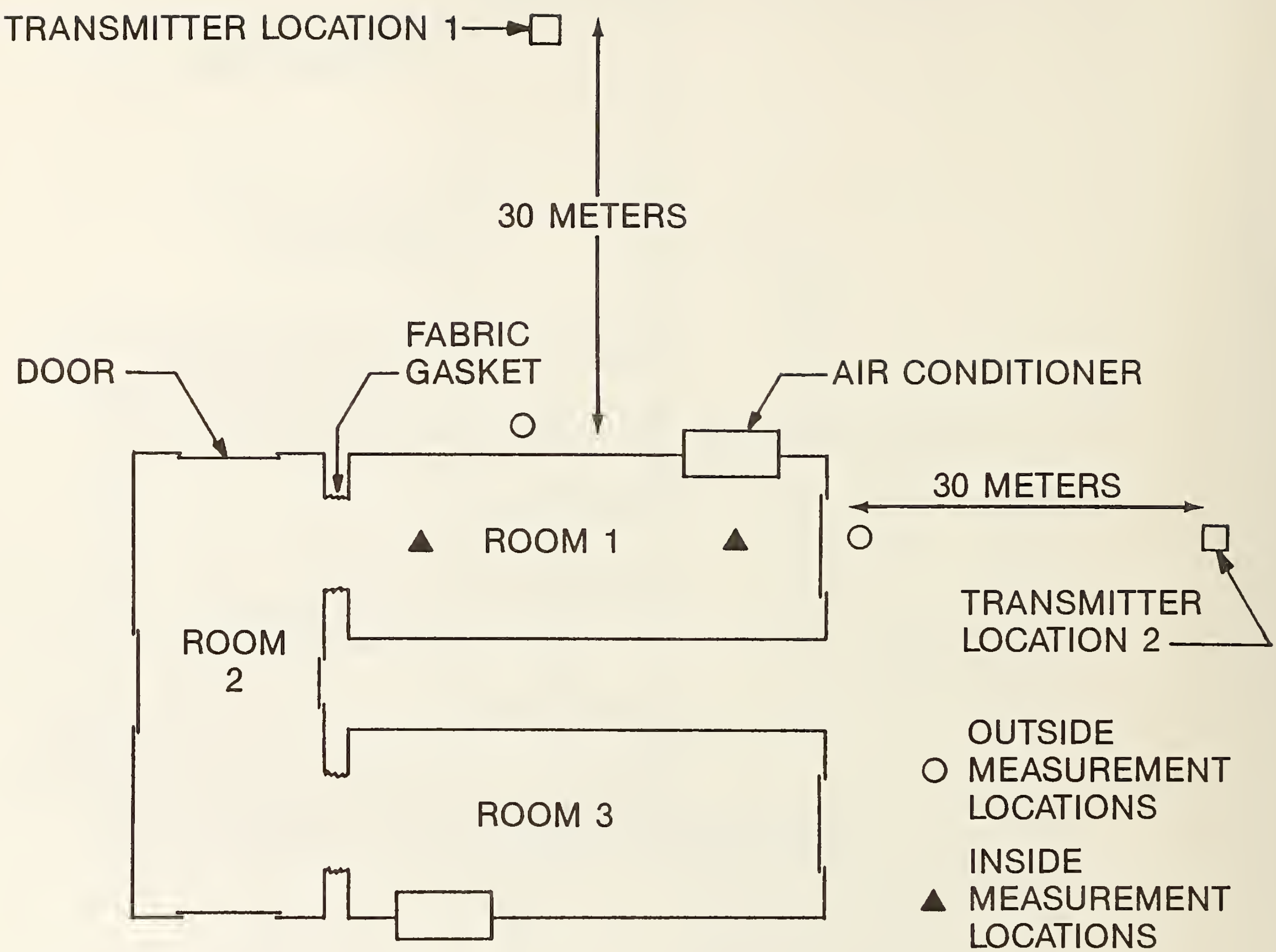


Figure 6.8. Building layout of training module at NTEC. The physical arrangement of the three attached buildings are shown along with the two transmitter locations (boxes). Air conditioning units are labeled "A.C.". A rubber shroud or gasket is used to attach the rooms together.

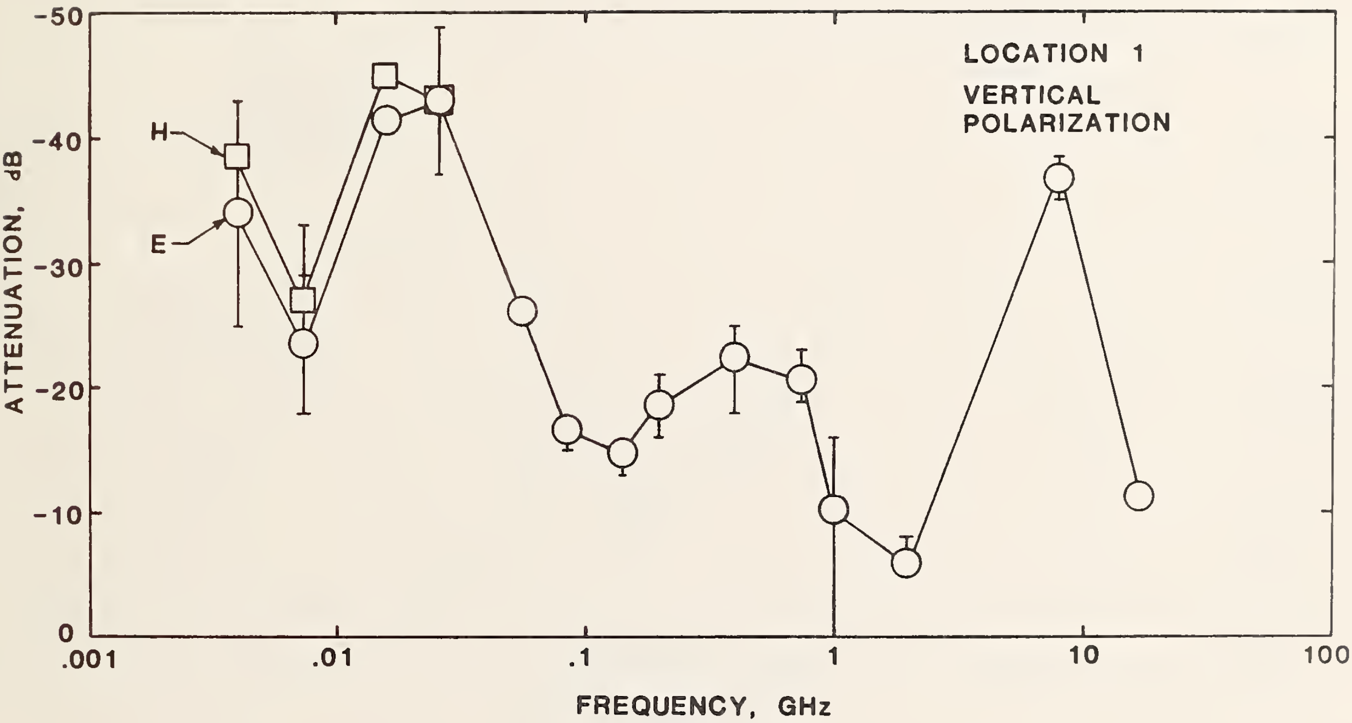


Figure 6.9. Building attenuation versus frequency of training module at NTEC. This graph shows the data for transmitter location 1 with vertical polarization launched. The circles represent average electric field attenuation, while the average magnetic field attenuation is shown as a square. The limit bars represent the highest and lowest attenuation observed at each frequency.

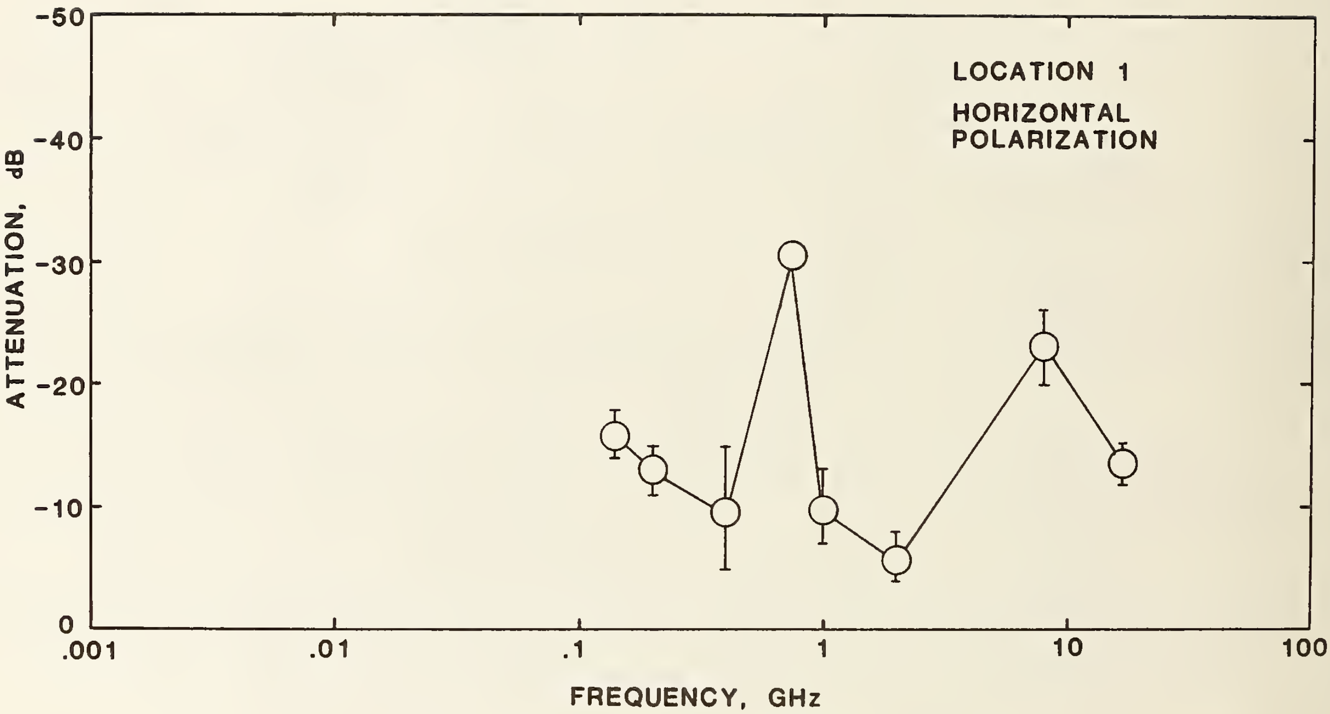


Figure 6.10. Building attenuation versus frequency of training module at NTEC. This graph shows the data for transmitter location 1 with horizontal polarization launched. The circles represent the average electric field attenuation, while the limit bars represent the highest and lowest attenuations observed at each frequency.

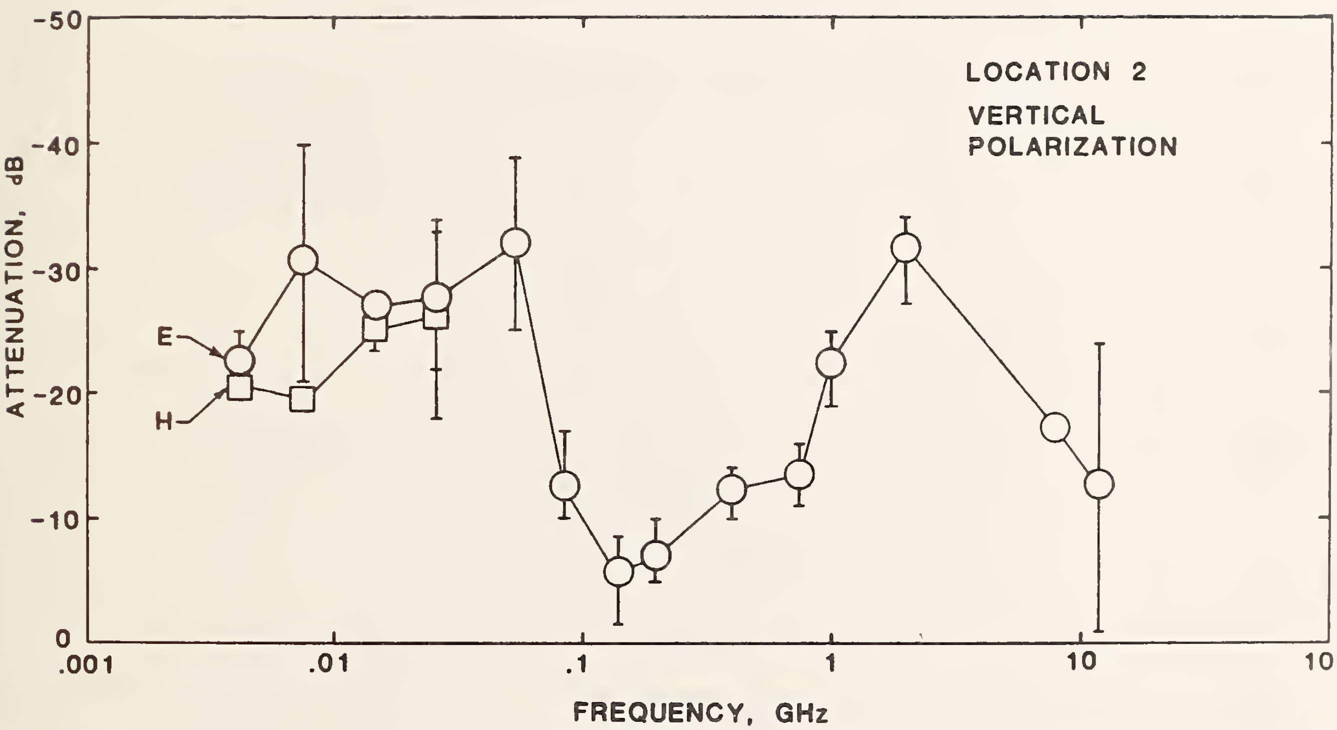


Figure 6.11. Building attenuation versus frequency of training module at NTEC. This graph shows the data for transmitter location 2 with vertical polarization launched. The circles represent average electric field attenuation, while the average magnetic field attenuation is shown as a square. The limit bars represent the highest and lowest attenuations observed at each frequency.

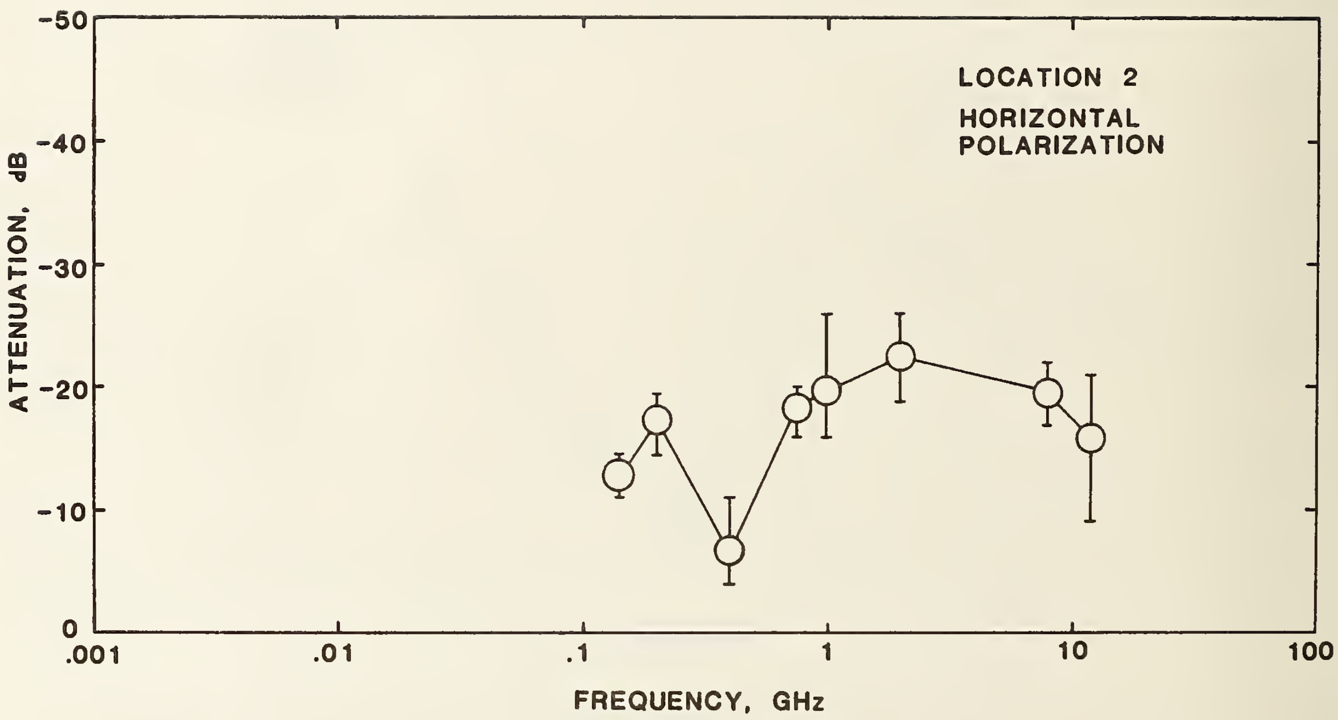


Figure 6.12. Building attenuation versus frequency of training module at NTEC. This graph shows the data for transmitter location 2 with horizontal polarization launched. The circles represent the average electric field attenuation, while the limit bars represent the highest and lowest attenuation observed at each frequency.

7. COMPUTER RESULTS AND CONCLUSIONS

7.1 Comparison With Experimental Results

To test the validity of the computer calculation of building attenuation, the program was run for a building that had been measured experimentally for electromagnetic shielding effectiveness. The structure chosen for comparison was the training module located at the Naval Training Equipment Center (NTEC) in Orlando, Florida. The experimental measurements of shielding effectiveness are already described in Section 6.2 of this report.

To input data into the computer program, the training module floor plan shown in figure 6.8, wa redrawn, as shown in figure 7.1, to show the details of the walls, windows, and doors, based on observations made at the measurement site. Some assumptions and compromises were also made so that the building could be properly modelled by the computer. Looking at figure 6.8, the main questions were: 1) how to account for the rubber gaskets between the modules. The first question was solved by considering the open space between the modules as a fourth room, or room 4 as shown on figure 7.1. The walls were taken as MØ1 ("NULL MATERIAL") and windows of material MØ1 with a metal frame (M12 (STEEL)) were added on the three ends of the room. The "T" shape of ROOM Ø4 was created by using two rectangular shaped volumes and giving them the same name. For example, look at the last two data entry lines of the first table in figure 7.2(a) and notice that room 4 has two ceiling sections, one with dimensions 6.1Ø x 1.22 m and the second with 6.1Ø x Ø.15 m. The three windows, DA, DB and DB, shown in the figure were added so that the input resonance condition described in Section 3.5 would be taken into account.

The air conditioners (question 2) were considered closed doors in the model ("DC" on the diagram). The rubber gaskets (question 3) were modelled by ignoring them. They should have no shielding effectiveness for electromagnetic radiation.

When the experimental measurements were made on the shelter, some of the exterior doors were left open to provide ventilation for the equipment from the hot and humid conditions. Those doors, marked "DO" in figure 7.1, are modelled as "open" for the computer program so that the calculations can be properly compared with the experimental results. (Experimentally it was found that opening the doors had less than a 2 dB effect on the measurements. Since equipment failure was experienced with the doors closed, and since the experimental uncertainty was typically greater than 2 dB, the engineers in charge of the measurement made the decision to leave those doors open.)

The computer print-out for the calculation is shown in figure 7.2 (a-d). The first three tables list the wall data file B2Ø4W, the hole data file B2Ø4H, and hole types data file B204T. The next thirteen tables list the attenuation of each room (1-4) for each direction of input (1-5) for frequencies in the range of 1.Ø MHz to 1Ø GHz. In figures 7.3-7.6, the experimental data (open circles and squares) is compared to the calculated data (closed triangles) where transmitter location 1 corresponds to computer direction D1, and transmitter location 2 corresponds to computer direction D2. Since experimental measurements were made only for room 1, the calculated data is displayed for just room 1.

By symmetry, a field projected from direction D3 should yield identical results with a field projected from D1. This is evident in the attenuation tables of figure 7.2 (a-d) where the column corresponding to D3 is identical with the results shown in the first column, D1. The fourth column, D4, is -6∅ dB at all frequencies and for all rooms. This is caused by the fact that the wall facing direction D4 has no openings which will allow penetration. Since that wall "shadows" all the other rooms, and since the computer model does not include external diffraction around corners, all of the rooms will have a -6∅ dB attenuation factor for direction D4. In figure 7.7, the room and door resonances of the test structure are shown. In the ∅.1 GHz range the room resonance and door resonance for door "DO" dominate and drop the attenuation factor to zero. (Where the computer model calculated gain for a room, the attenuation factor was taken to be zero.) This effect was equally strong for both directions D1 and D2 and can be seen in figures 7.3 and 7.4. At around 1.∅ GHz, the dominant resonant effect is due to door "DB". Since this door is only illuminated from direction D1, the calculated attenuation at 1.∅ GHz is reduced for direction D1 (fig. 7.3) but not for direction D2 (fig. 7.4). This is in good agreement with the corresponding experimental measurements for those directions.

Overall, the fit between the calculated and experimental results are in good agreement. Given just the calculated data, it would be possible to estimate the shielding effectiveness of the training modules.

7.2 Recommendation for Further Work

One area in the model that could use further development is in the resonance calculations. The present approach essentially turns the resonances "on" or "off" and does not use any sophisticated techniques to properly weight the resonant effects. It should be possible to incorporate more advanced resonant models into the program. That task should be addressed in future work.

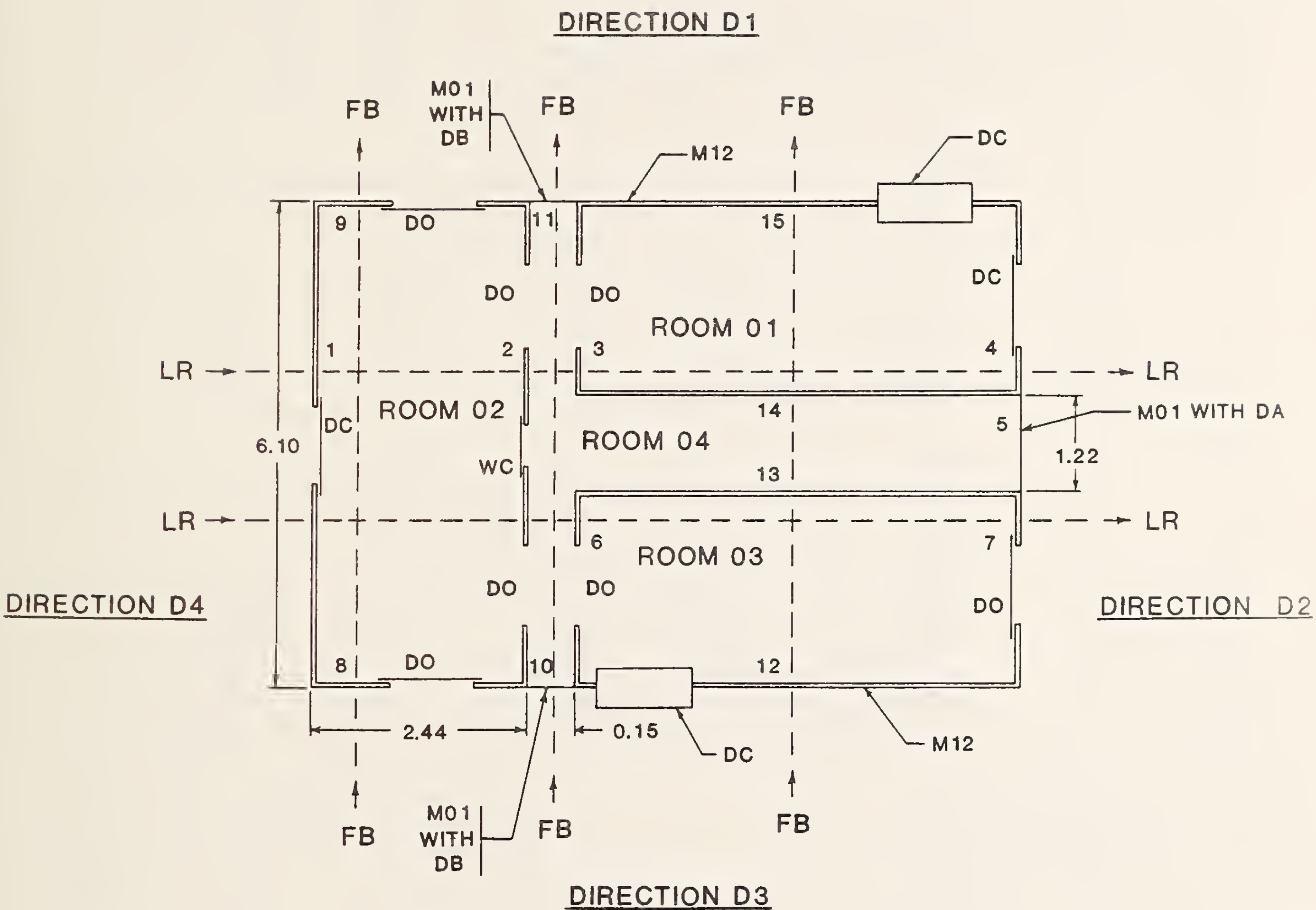


Figure 7.1. Building layout of training modules used for comparison of calculated versus experimental data. Notice that the air gap between the modules is considered "Room Ø4" for the computer model.

```
ENTER BUILDING IDENTIFICATION  (E.G. '101')
    (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)
? '204'
BUILDING IDENTIFICATION ENTERED AS '204'
ENTER NUMBER OF ROOMS IN BUILDING
? 4
```

WALL IDENTIFICATION			WALL PARAMETERS			
DIR	FROM	TO	MATERIAL	HEIGHT	WIDTH	THICKNESS
LR	D4	02	M12	2.44	6.10	.20
LR	02	04	M12	2.44	6.10	.20
LR	04	01	M12	2.44	2.44	.20
LR	01	D2	M12	2.44	2.44	.20
LR	04	D2	M01	2.44	1.22	.20
LR	04	03	M12	2.44	2.44	.20
LR	03	D2	M12	2.44	2.44	.20
FB	D3	02	M12	2.44	2.44	.20
FB	02	D1	M12	2.44	2.44	.20
FB	D3	04	M01	2.44	.15	.20
FB	04	D1	M01	2.44	.15	.20
FB	D3	03	M12	2.44	6.10	.20
FB	03	04	M12	2.44	6.10	.20
FB	04	01	M12	2.44	6.10	.20
FB	01	D1	M12	2.44	6.10	.20
UD	D5	01	M12	6.10	2.44	.20
UD	D5	02	M12	6.10	2.44	.20
UD	D5	03	M12	6.10	2.44	.20
UD	D5	04	M01	6.10	1.22	1.00
UD	D5	04	M01	6.10	.15	1.00
UD	01	D6	M12	6.10	2.44	.20
UD	02	D6	M12	6.10	2.44	.20
UD	03	D6	M12	6.10	2.44	.20
UD	04	D6	M01	6.10	1.22	1.00
UD	04	D6	M01	6.10	.15	1.00

DOOR AND WINDOW LOCATIONS

ID	WALL IDENTIFICATION DIRECTION	FROM	TO
DC	LR	D4	02
DO	LR	02	04
DO	LR	04	01
DC	LR	01	D2
WC	LR	02	04
DA	LR	04	D2
DO	LR	02	04
DO	LR	04	03
DO	LR	03	D2
DC	FB	D3	03
DO	FB	D3	02
DO	FB	02	D1
DE	FB	D3	04
DB	FB	04	D1
DC	FB	01	D1

Figure 7.2(a). Computer print-out of calculations of electromagnetic shielding effectiveness for building shown in figure 7.1.

DOOR AND WINDOW PARAMETERS

ID	MATERIAL	FRAME MATERIAL	HEIGHT	WIDTH	LAYER THICKNESS	DISTANCE ABOVE FLR
DA	M01	M12	2.44	1.22	.20	0.00
DB	M01	M12	2.44	.15	.20	0.00
DC	M12	M12	1.76	1.22	.20	0.00
DO	M01	M12	1.76	1.22	.20	0.00
WC	M12	M12	.61	.46	.20	0.00
WA	M10	M12	1.76	1.22	.20	0.00

ATTENUATION AT A FREQUENCY OF 1.000E+06 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-32.42	-23.59	-32.42	-60.00	-28.73
2	-6.96	-23.31	-6.96	-60.00	-28.45
3	-35.71	-7.48	-35.71	-60.00	-32.03
4	-12.73	-3.91	-12.73	-60.00	-9.05

ATTENUATION AT A FREQUENCY OF 2.000E+06 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-32.42	-23.59	-32.42	-60.00	-28.73
2	-6.96	-23.31	-6.96	-60.00	-28.45
3	-35.71	-7.48	-35.71	-60.00	-32.03
4	-12.73	-3.91	-12.73	-60.00	-9.05

ATTENUATION AT A FREQUENCY OF 5.000E+06 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-32.42	-23.59	-32.42	-60.00	-28.73
2	-6.96	-23.31	-6.96	-60.00	-28.45
3	-35.71	-7.48	-35.71	-60.00	-32.03
4	-12.73	-3.91	-12.73	-60.00	-9.05

ATTENUATION AT A FREQUENCY OF 1.000E+07 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-32.42	-23.59	-32.42	-60.00	-28.73
2	-6.96	-23.31	-6.96	-60.00	-28.45
3	-35.71	-7.48	-35.71	-60.00	-32.03
4	-12.73	-3.91	-12.73	-60.00	-9.05

Figure 7.2(b). Computer print-out of calculations of electromagnetic shielding effectiveness for building shown in figure 7.1.

ATTENUATION AT A FREQUENCY OF 2.000E+07 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-32.42	-23.59	-32.42	-60.00	-28.73
2	-6.96	-23.31	-6.96	-60.00	-28.45
3	-35.71	-7.48	-35.71	-60.00	-32.03
4	-12.73	-3.91	-12.73	-60.00	-9.05

ATTENUATION AT A FREQUENCY OF 5.000E+07 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-32.42	-23.59	-32.42	-60.00	-28.73
2	-6.96	-23.31	-6.96	-60.00	-28.45
3	-35.71	-7.48	-35.71	-60.00	-32.03
4	-12.73	-3.91	-12.73	-60.00	-9.05

ATTENUATION AT A FREQUENCY OF 1.000E+08 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	5.65	10.41	5.65	-60.00	-15.70
2	9.40	7.51	9.40	-60.00	-18.60
3	2.08	12.18	2.08	-60.00	-19.27
4	6.84	11.60	6.84	-60.00	-14.51

ATTENUATION AT A FREQUENCY OF 2.000E+08 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	5.65	10.41	5.65	-60.00	-15.70
2	9.40	7.51	9.40	-60.00	-18.60
3	2.08	12.18	2.08	-60.00	-19.27
4	6.84	11.60	6.84	-60.00	-14.51

ATTENUATION AT A FREQUENCY OF 5.000E+08 HZ

ROOMS	DIRECTIONS 1	2	3	4	5
1	-10.11	-21.00	-10.11	-60.00	-26.14
2	-5.49	-21.78	-5.49	-60.00	-26.92
3	-14.39	-7.39	-14.39	-60.00	-30.42
4	8.51	-2.37	8.51	-60.00	-7.51

Figure 7.2(c). Computer print-out of calculations of electromagnetic shielding effectiveness for building shown in figure 7.1.

ATTENUATION AT A FREQUENCY OF 1.000E+09 HZ

ROOMS	1	2	3	4	5
1	-10.11	-21.00	-10.11	-60.00	-26.14
2	-5.49	-21.78	-5.49	-60.00	-26.92
3	-14.39	-7.39	-14.39	-60.00	-30.42
4	8.51	-2.37	8.51	-60.00	-7.51

ATTENUATION AT A FREQUENCY OF 2.000E+09 HZ

ROOMS	1	2	3	4	5
1	-10.11	-21.00	-10.11	-60.00	-26.14
2	-5.49	-21.78	-5.49	-60.00	-26.92
3	-14.39	-7.39	-14.39	-60.00	-30.42
4	8.51	-2.37	8.51	-60.00	-7.51

ATTENUATION AT A FREQUENCY OF 5.000E+09 HZ

ROOMS	1	2	3	4	5
1	-32.42	-23.59	-32.42	-60.00	-28.73
2	-6.96	-23.31	-6.96	-60.00	-28.45
3	-35.71	-7.48	-35.71	-60.00	-32.03
4	-12.73	-3.91	-12.73	-60.00	-9.05

ATTENUATION AT A FREQUENCY OF 1.000E+10 HZ

ROOMS	1	2	3	4	5
1	-32.42	-23.59	-32.42	-60.00	-28.73
2	-6.96	-23.31	-6.96	-60.00	-28.45
3	-35.71	-7.48	-35.71	-60.00	-32.03
4	-12.73	-3.91	-12.73	-60.00	-9.05

Figure 7.2(d). Computer print-out of calculations of electromagnetic shielding effectiveness for building shown in figure 7.1.

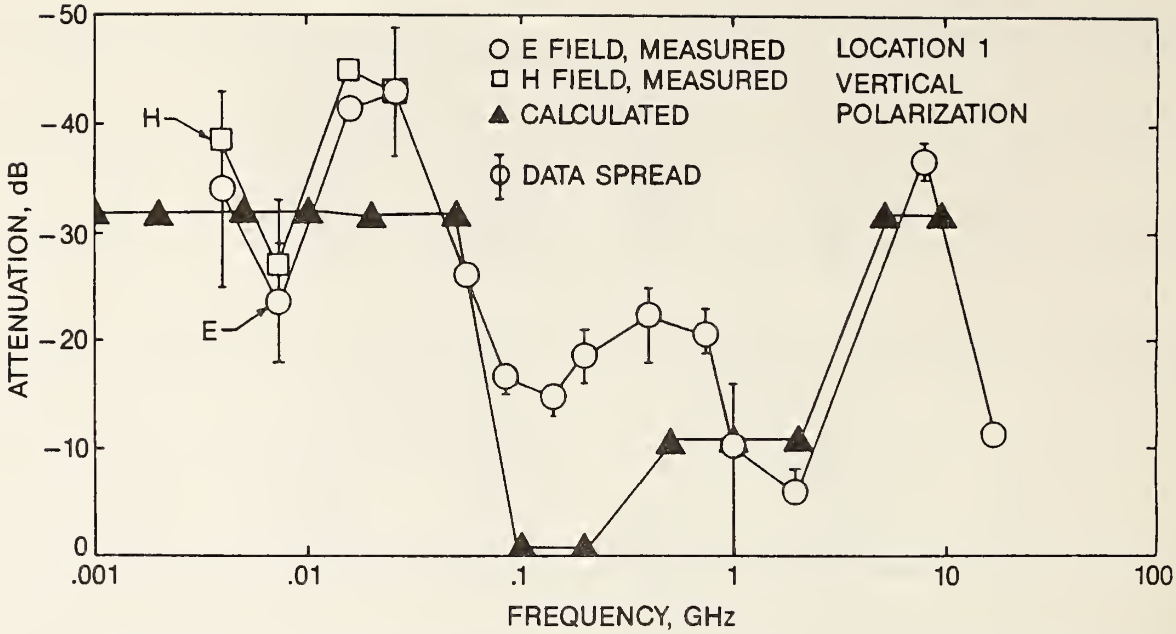


FIGURE 7.3 TRANSMITTER LOCATION 1, VERTICAL POLARIZATION DATA

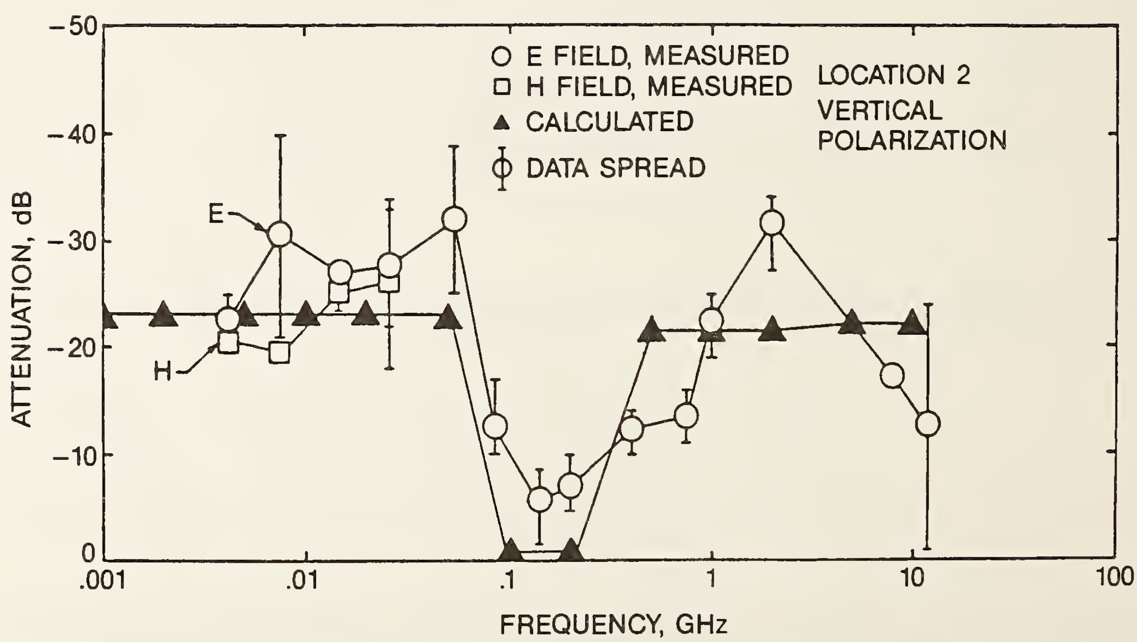


FIGURE 7.4 TRANSMITTER LOCATION 2, VERTICAL POLARIZATION DATA

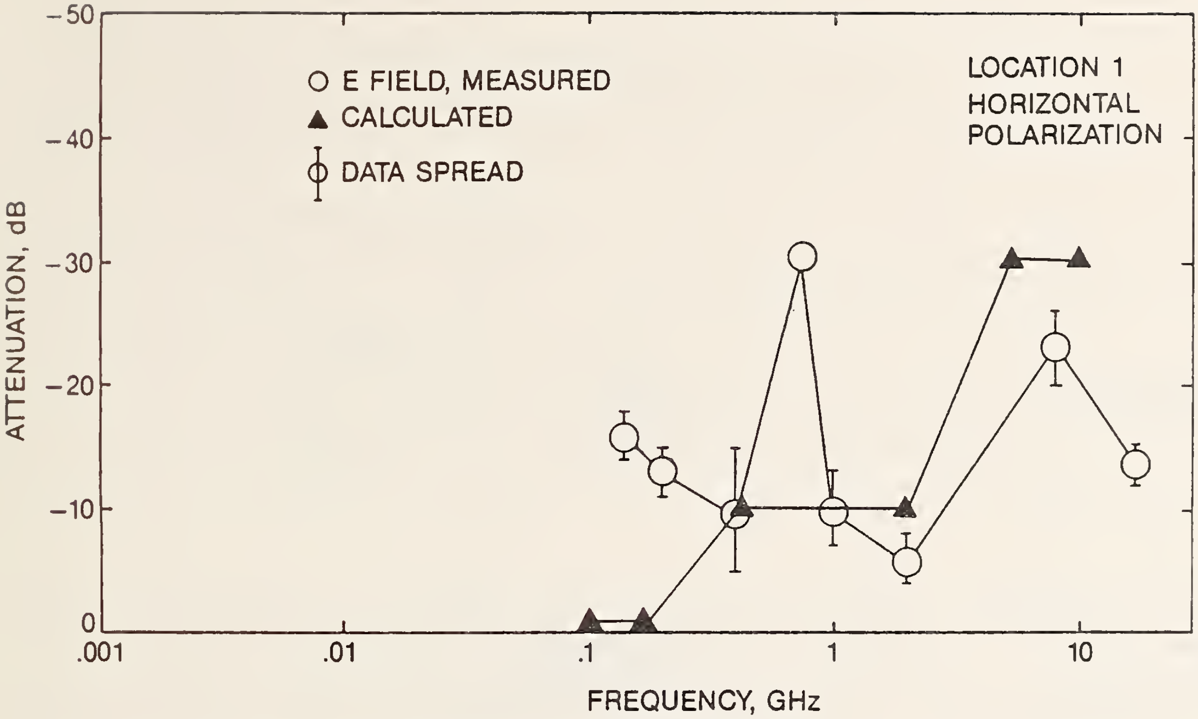


FIGURE 7.5 TRANSMITTER LOCATION 1, HORIZONTAL POLARIZATION DATA

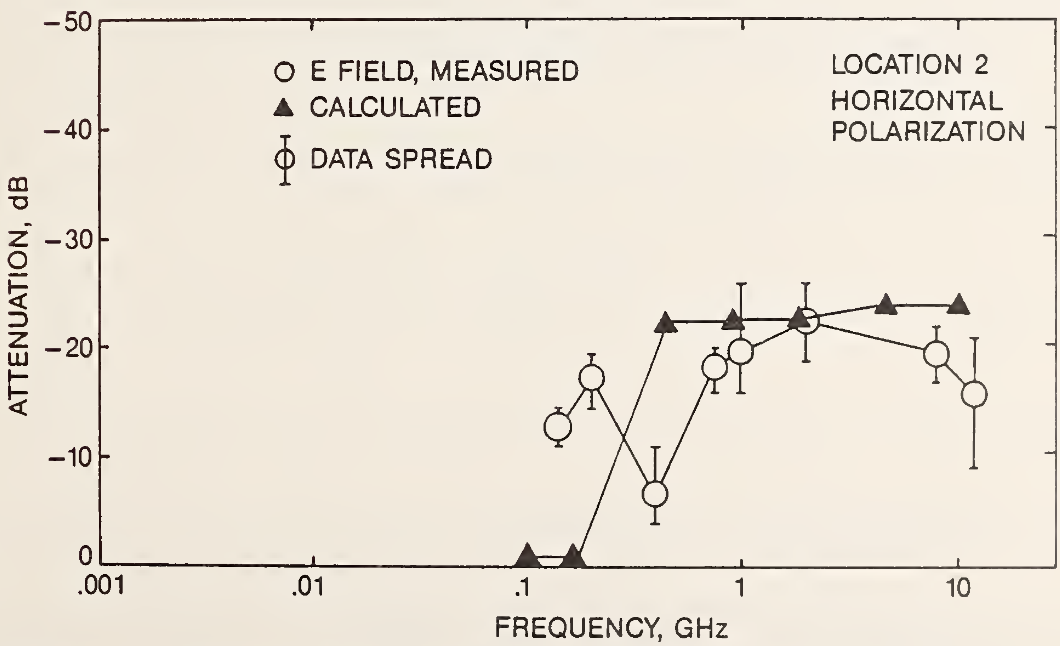


FIGURE 7.6 TRANSMITTER LOCATION 2, HORIZONTAL POLARIZATION DATA

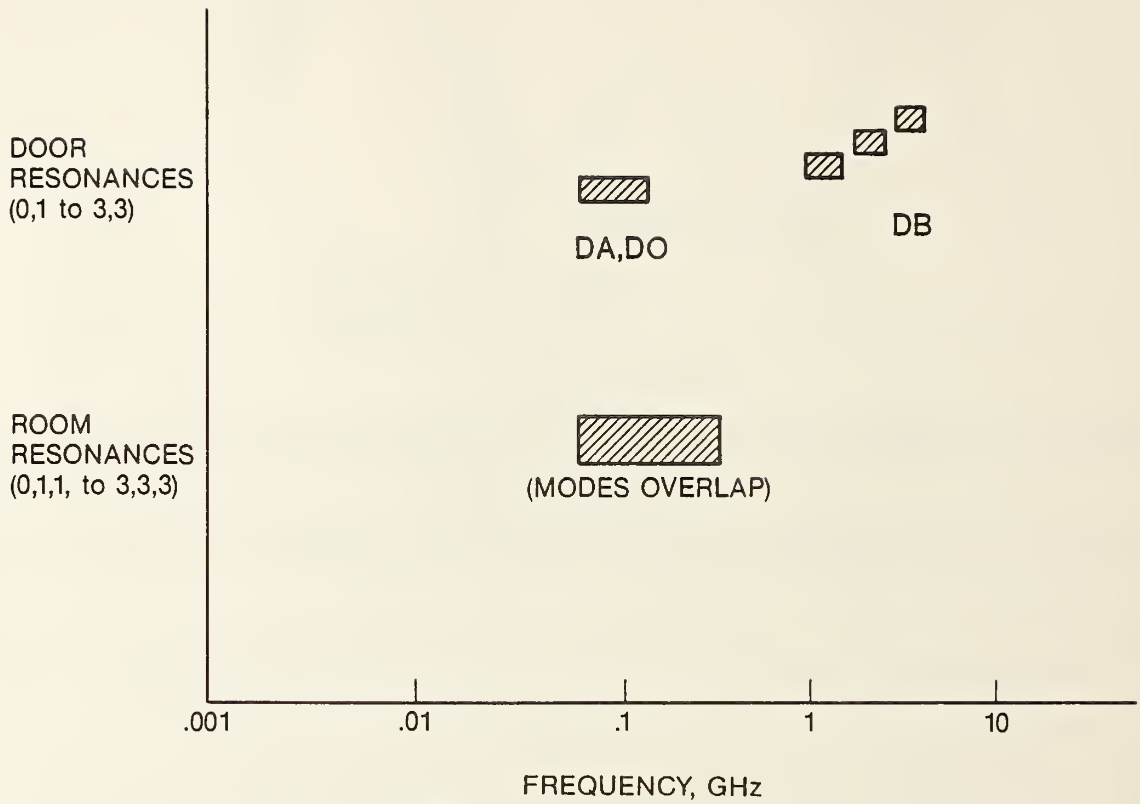


FIGURE 7.7 TEST STRUCTURE ROOM AND DOOR RESONANCES

8. REFERENCES

[1] Cox, D. C.; Murray, R. R.; Norris, A. W. Measurements of 800 MHz radio transmission into buildings with metallic walls. To be published in Bell System Technical Journal.

[2] Rice, L. P. Radio transmission into buildings on 35 and 150 MHz. Bell System Technical Journal 38: 197-210; 1959 January.

[3] Smith, A. A., Jr. Attenuation of electric and magnetic fields by buildings. IEEE Trans. Electromag. Compat. EMC-20(3): 411-418; 1978 August.

[4] Brennan, C. M.; Lambert, C. C.; Conner, C. G.; Roberts, G. F.; Flannery, W. T. EM shielding of building materials. Electro-Mechanics Co., Austin, Texas. Rome Air Development Center final report RADC-TR-67-446; 1968 February. 183 p. (AD-667-546).

[5] Garrett, K. S.; Flannery, W. T.; Osburn, J. D.; Brennan, C. M.; Flory, R. K. A study of the shielding effectiveness of building materials. Electro-Mechanics Co., Austin, Texas. Rome Air Development Center final report RADC-TR-66-178; 1966 May. 382 p. (AD-485-027/7ST).

[6] Johnk, Carl T.A. Engineering electromagnetic fields and waves. New York: John Wiley and Sons; 1975. 655 p.

[7] Denny, H. W.; Holland, L. D.; Robinette, S. L.; Woody, J. A. Grounding, bonding, and shielding practices and procedures for electronic equipments and facilities. Vol. I: Fundamental considerations. Engineering Experiment Station, Georgia Institute of Technology. Federal Aviation Administration final report FAA-RD-75-215; 1975 December. 399 p. (AD-A022-332).

[8] Campi, M. Survey and review of building shielding to electromagnetic waves from EMP. Harry Diamond Laboratories, Adelphi, Md. Final report HDL-TM-78-23; 1978 November. 109 p. (AD-B034-726L).

[9] Jakubec, L. G.; Ohta, H. H. Proposed specifications for electromagnetic shielding of enclosures and buildings. Genistron, Incorporated, Los Angeles, CA. Phase I final report to U.S. Naval Civil Engineering Laboratory on contract NBy-32220; 1963 July. (AD-417-699).

[10] Jarva, W. Shielding efficiency calculation methods for screening, waveguide ventilation panels, and other perforated electromagnetic shields. Proc. of the Seventh Conf. on Radio Interference Reduction and Electronic Compatibility. 1961 November; Armour Research Foundation of Ill. Inst. of Technology, and Professional Group on Radio Frequency Interference of the Inst. of Radio Engineers.

[11] MacFarlane, G. G. Surface impedance of an infinite parallel-wire grid at oblique angles of incidence. Proc. IEE 93(IIIA): 1523-1527; 1946.

[12] Hill, D. A.; Wait, J. R. Scattering of a transient plane wave by a periodic grating. Radio Sci. 11(6): 1003-1009; 1971 November.

[13] Hill, D. A.; Wait, J. R. Electromagnetic scattering of an arbitrary plane wave by two nonintersecting perpendicular wire grids. Can. J. Phys. 52(3): 227-237; 1974.

[14] Hill, D. A.; Wait, J. R. Electromagnetic scattering of an arbitrary plane wave by a wire mesh with bonded junctions. Can. J. Phys. 54(4): 353-361; 1976.

[15] Kontorovich, M. I.; Petrun'kin, V. Yu.; Yesephkina, N. A.; Astrakhan, M. I. The coefficient of reflection of a plane electromagnetic wave from a plane wire mesh. Radio Eng. Elect. Phys. 7: 222-231; 1962.

[16] Astrakhan, M. I. Reflecting and screening properties of plane wire grids. Radio Engineering. 23(1): 76-83; 1968.

[17] James, W. L. Dielectric properties of wood and hardboard: variation with temperature, frequency, moisture content, and grain orientation. USDA Forest Service, Forest Products Laboratory, Madison, Wisc. USDA Forest Service Research Paper FPL 245; 1975. 32 p. (AD-A015-654).

[18] Johnson, M. K. Dielectric constant and loss tangent determination for various dielectric materials to be used at the Trestle site. Braddock, Dunn, and McDonald, Inc., Albuquerque, New Mexico. Prepared for U.S. Air Force Trestle program. Report BDM/A-83-74-TR; 1974 July 4. 88 p. (AD-B008-055L).

APPENDIX 9.1

TEST PLAN

BUILDING ATTENUATION MEASUREMENTS

A.1 Purpose

This plan outlines a method for measuring the attenuation of buildings over all or part of the frequency range covered in this report: 1Ø kHz - 1Ø GHz. In order to compare measured attenuation data with computed data (generated by the computer program developed for this report), the building must be square or rectangular, or made up of adjoining squares or rectangles; and the test signals must be perpendicularly incident on the building walls.

A.2 Frequency Coverage

Apart from limitations imposed by the availability of portable sources and launching antennas, the choice of measurement frequencies may be determined by factors such as

- regions of low attenuation in the computed data.
- the frequencies of strong ambient signals.
- frequencies at which a given device is known to have highest susceptibility.

If there are not worrisome frequencies, the project engineer can make the choice of measurement frequencies over the 1Ø kHz - 1Ø GHz range.

Table A-1 lists the frequencies at which an NBS team made building attenuation measurements on an Army installation. Other columns in the table indicate distance increments between locations on the measurement grid, the type of field to be measured and the type to be launched, and the electric field polarization. A measurement grid is shown in figure A.1 for a single-room building or for a single room within a building, though the size, shape, and construction of a particular building may permit it to be characterized by many fewer measurement locations (see section A.3).

A.3 Number and location of measurements

A.3.1 For a free-standing, rectangular building with a single room, a field will be launched perpendicular to all four faces of the building in turn. Where many rooms exist, only some of which are of interest, or the general building geometry dictates, some of the four faces may not be used.

A.3.2 The exact pattern of measurement location is determined by the frequency and is adaptive. (Ref. figure A.1).

Table A.1. Tabulation of the frequencies and signal properties for a set of electromagnetic field attenuation measurements.

Frequency	Δ, Meters[1,2]	Type of Measurement	Field Launched	Polarization[3]
180 kHz	8	E & H	Primarily H	Vert.
3.5 MHz	8	E & H	E X H	"
7 MHz	4	E	E X H	"
14 MHz	4	E	E X H	"
28 MHz	2	E	E X H	"
54 MHz	2	E	E X H	"
140 MHz	1*	E	E X H	"
200 MHz	1*	E	E X H	"
400 MHz	1*	E	E X H	"
750 MHz	1*	E	E X H	"
1000 MHz	1*	E	E X H	Circular or Horiz.
2000 MHz	1*	E	E X H	"
4000 MHz	1*	E	E X H	"
8000 MHz	1*	E	E X H	"
12000 MHz	1*	E	E X H	"
18000 MHz	1*	E	E X H	"

Notes

1. * Actual position will be varied ± 1/4 meter to achieve highest reading.
2. Δ is spacing of locations for sequence 3 and above.
3. Circular polarization will be used above 1 GHz if adequate signals are received inside building. If signal levels are too low a high gain horizontally polarized launching antenna will be used.

A.3.2.1 First, the incident field is measured 5 meters from the face of the building at three locations as indicated.

A.3.2.2 Next, a 1 meter by 1 meter grid is established in the area to be measured. A line, two meters inside the front wall is measured every meter to within 2 meters of the side walls. This will be called sequence A.

A.3.2.3 The highest reading is noted and a line of points Δ meters apart (see Table A.1), perpendicular to the first line is measured, until a total-field measurement is obtained which is equal to or less than the lowest reading obtained in sequence A. This will be called sequence B. Note, however, if this line of measurements is within ± 3 meters of the center line of the building, this sequence may be eliminated.

A.3.2.4 A line of points Δ meters apart is now measured along the center line of the building to the center of the room. This is called sequence C.

A.3.2.5 If no other faces of the building are to be excited which are perpendicular to the first face, then a last sequence D across the middle of the room will be taken at spacings of Δ meters.

A.3.2.6 If L/Δ or $W/\Delta \leq 5$, (where L & W are dimensions of the room being measured) then at least 5 measurement points will be taken in sequences B, C, and D. The nearest full meter value for Δ will be chosen which will result in at least 5 measurement points within the room's dimensions.

A.3.2.7 All readings will be taken at a height of 1.5 meters except those at 140 MHz when a second set will be taken at a height of 1 meter.

A.3.2.8 Data will be recorded for the X, Y, and Z orthogonal components, plus the equivalent vector sum.

A.3.3 In buildings with a large door opening, data will also be taken with the door open at the frequency where the vertical dimension of the opening is $\lambda/2$, and at 7 MHz, 400 MHz, and 4 GHz.

Again, we emphasize that the test plan may be modified for each test site by the engineers in charge to conform to any constraints imposed by time or the physical layout of the structure. Although the test plan recommends that measurements be grouped in four locations (fig. A.1), the engineers in charge may decide that three (or in some cases, two) groups are sufficient to determine the shielding effectiveness of the buildings.

A.4 Data Presentation

Attenuation measurements can be summarized in tables and also plotted. The table for each building, room, and transmitter location can list the frequencies of the test, the mean values of attenuation measured, and uncertainties of the measurement (one standard deviation) shown in parentheses. The mean value of attenuation is determined by averaging all grid point measurements in a particular building or room, for a particular frequency for each transmitter location:

$$\bar{X} = \frac{1}{N} \Sigma x_i$$

where $\bar{X}$ is the sample mean, N is the number of grid points measured, and x_i represents the individual attenuations at each measured grid point. The standard deviation is defined as the square root of the sample variance:

$$\sigma = \sqrt{s^2}$$

where

$$s^2 = \frac{1}{N-1}\Sigma(x_i-\bar{X})^2$$

and where σ is the standard deviation, s^2 is the sample variance, and x_i and $\bar{X}$ are already defined. The graphs show the mean attenuation at each measurement frequency, with the one-standard-deviation limits as error bars.

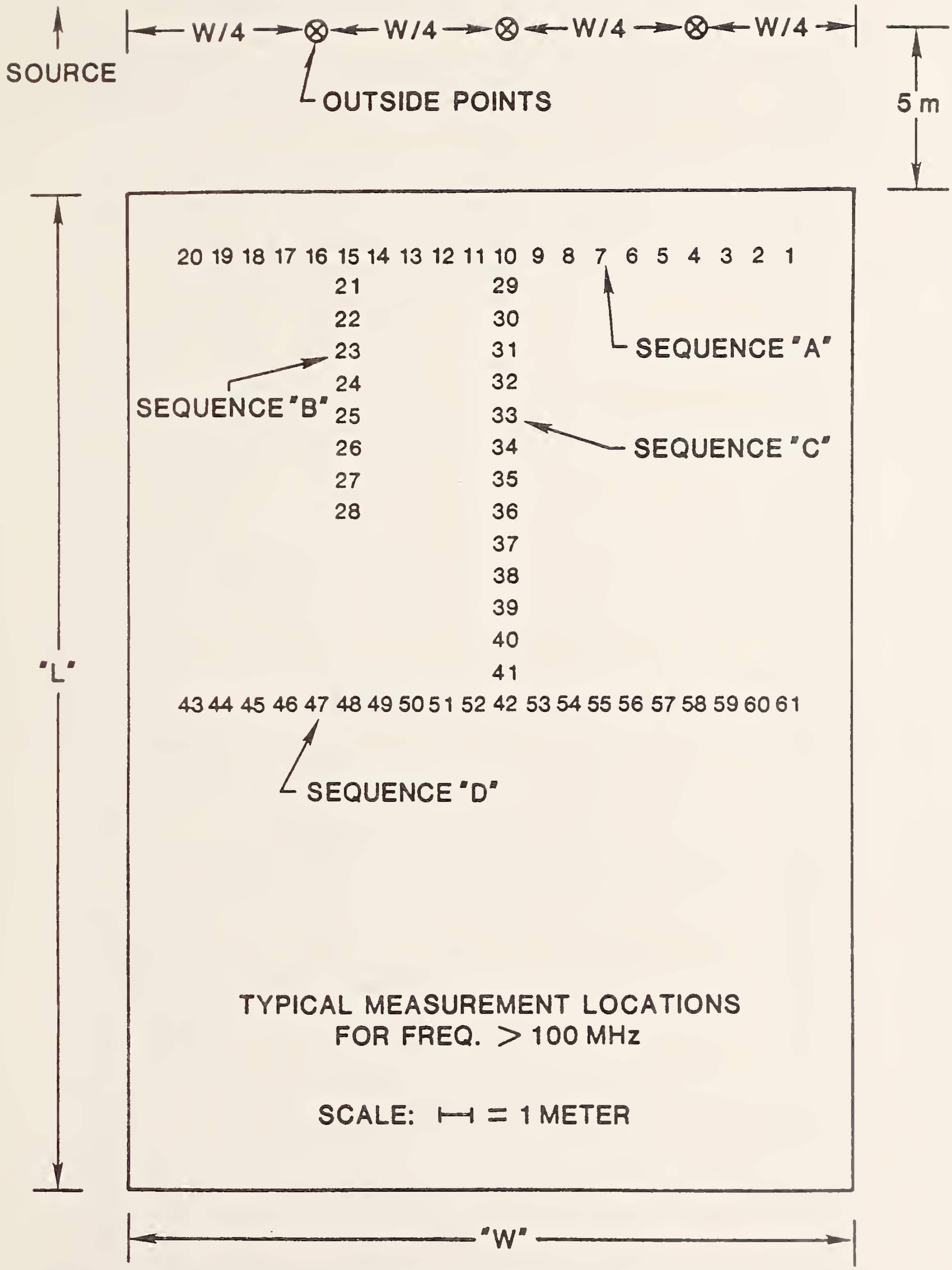


Figure A.1 Typical measurement locations for frequencies greater than 100 MHz

Appendix 9.2 Listing of Computer Program SMATDB

```
      PROGRAM SMATDB                                                    SMATDB       1
C**  INPUT MATERIAL ATTENUATION AND REFLECTION COEFFICIENT INTO         SMATDB       2
C**     ARRAYS AND THEN STORE DATA IN A PERMANENT FILE*                 SMATDB       3
**                                                                      SMATDB       4
***********************************************************************COMM         1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                     ***COMM         2
***********************************************************************COMM         3
      INTEGER MMAX                                                      COMM         4
      PARAMETER (MMAX=100)                                              COMM         5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM         6
     $              MFREQ(MMAX,7), MERR, MTOT                           COMM         7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                              COMM         8
      INTEGER MTOT, MERR                                                COMM         9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                COMM        10
      CHARACTER * 3  MAT                                                COMM        11
      CHARACTER * 70 MATDESC                                            COMM        12
***********************************************************************COMM        13
***********************************************************************COMM        14
      INTEGER R,C, COMMAND                                              SMATDB       6
      CHARACTER * 3 MATID                                               SMATDB       7
C*  INITIALIZE ARRAYS                                                   SMATDB       8
      DATA MAT / 100 * '   ' /                                          SMATDB       9
      DATA MATDESC / 100 * '   ' /                                      SMATDB      10
      DATA MFREQ / 700 * 0.0 /                                          SMATDB      11
      DATA MATTEN / 700 * 0.0 /                                         SMATDB      12
      DATA QA / 100 * 0.0 /                                             SMATDB      13
      DATA MRCOEF / 700 * 0.0 /                                         SMATDB      14
      DATA QR / 100 * 0.0 /                                             SMATDB      15
C*  ENTER COMMANDS                                                      SMATDB      16
      PRINT *, '(1) CREATE NEW DATABASE  (2) ADD TO EXISTING DATA ',    SMATDB      17
     Z       'BASE'                                                     SMATDB      18
      READ *, COMMAND                                                   SMATDB      19
      IF (COMMAND .EQ. 2) THEN                                          SMATDB      20
         CALL LMATTER                                                   SMATDB      21
         IF (MERR .NE. 0) CALL ERROR(5)                                 SMATDB      22
         ENDIF                                                          SMATDB      23
10    PRINT*,'(1) NEXT DATA ENTRY  (2) CHANGE  (3) DISPLAY  (4) ABORT', SMATDB      24
     $       '  (5) QUIT'                                               SMATDB      25
      READ *, COMMAND                                                   SMATDB      26
      PRINT *                                                           SMATDB      27
      IF (COMMAND .EQ. 1) THEN                                          SMATDB      28
         CALL NEXT                                                      SMATDB      29
      ELSE IF (COMMAND .EQ. 2 ) THEN                                    SMATDB      30
         CALL CHANGE                                                    SMATDB      31
      ELSE IF (COMMAND .EQ. 3 ) THEN                                    SMATDB      32
         CALL DISPLAY                                                   SMATDB      33
      ELSE IF (COMMAND .EQ. 4 ) THEN                                    SMATDB      34
        PRINT*                                                          SMATDB      35
        PRINT*,'PROGRAM ABORTED AT YOUR REQEST'                         SMATDB      36
        PRINT*                                                          SMATDB      37
        STOP                                                            SMATDB      38
      ELSE IF (COMMAND .EQ. 5 ) THEN                                    SMATDB      39
        CALL QUIT                                                       SMATDB      40
        STOP                                                            SMATDB      41
      ENDIF                                                             SMATDB      42
      GOTO 10                                                           SMATDB      43
      END                                                               SMATDB      44
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  C         NONE                UNUSED/*S*       INTEGER
  COMMAND    157B                                INTEGER
  MAT          0B   /MATC/                       CHAR*3         100
  MATDESC     36B   /MATC/                       CHAR*70        100
  MATID     NONE                UNUSED/*S*       CHAR*3
  MATTEN       0B   /MATN/                       REAL           700
  MERR      4374B   /MATN/                       INTEGER
  MFREQ     3100B   /MATN/                       REAL           700
  MRCOEF    1274B   /MATN/                       REAL           700
  MTOT      4375B   /MATN/                       INTEGER
  QA        2570B   /MATN/                       REAL           100
  QR        2734B   /MATN/                       REAL           100
  R         NONE                UNUSED/*S*       INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  MMAX    INTEGER                       100


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  CHANGE                   0      SUBROUTINE       LMATTER                   0      SUBROUTINE
  DISPLAY                  0      SUBROUTINE       NEXT                      0      SUBROUTINE
  ERROR                    1      SUBROUTINE       QUIT                      0      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10     24B                     37


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  SMATDB      5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH            160B =   112
  CM LABELLED COMMON LENGTH     5730B =  3032
  CM STORAGE USED              61000B = 25088
  COMPILE TIME                 0.081 SECONDS
```

```
                                                                              SMATDB   45
      SUBROUTINE NEXT                                                         SMATDB   46
*************************************************************************COMM     1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                       ***COMM     2
*************************************************************************COMM     3
      INTEGER MMAX                                                       COMM     4
      PARAMETER (MMAX=100)                                               COMM     5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM     6
     $              MFREQ(MMAX,7), MERR, MTOT                            COMM     7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                               COMM     8
      INTEGER MTOT, MERR                                                 COMM     9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                 COMM    10
      CHARACTER * 3  MAT                                                 COMM    11
      CHARACTER * 70 MATDESC                                             COMM    12
*************************************************************************COMM    13
*************************************************************************COMM    14
      INTEGER R,C, VAL, INDEX                                            SMATDB   48
      CHARACTER * 3 MATID                                                SMATDB   49
      LOGICAL ENTERED                                                    SMATDB   50
 10   PRINT *, 'MATERIAL I.D.?  (E.G.  ''M05'' OR ''M12'')'               SMATDB   51
      READ *, MATID                                                      SMATDB   52
      IF (MATID(1:1) .NE. 'M') THEN                                      SMATDB   53
         PRINT *, 'FIRST CHARACTER MUST BE AN M.   TRY AGAIN'            SMATDB   54
         GOTO 10                                                         SMATDB   55
         ENDIF                                                           SMATDB   56
C* CHECK IF THIS MATERIAL IS ALREADY ENTERED                              SMATDB   57
      ENTERED = .FALSE.                                                  SMATDB   58
      DO 20 R=1,MMAX                                                     SMATDB   59
      IF (MAT(R) .EQ. MATID) THEN                                        SMATDB   60
         ENTERED = .TRUE.                                                SMATDB   61
         INDEX  = R                                                      SMATDB   62
         ENDIF                                                           SMATDB   63
 20   CONTINUE                                                           SMATDB   64
C* ENTER NEW DATA IF MATERIAL NOT ALREADY ENTERED                         SMATDB   65
      IF (ENTERED) THEN                                                  SMATDB   66
         PRINT *, 'MATERIAL ',MATID,' ALREADY ENTERED'                   SMATDB   67
      ELSE                                                               SMATDB   68
         INDEX  = VAL ( MATID(2:3) )                                     SMATDB   69
      PRINT *,'INDEX:',INDEX                                             SMATDB   70
         MAT (INDEX) = MATID                                             SMATDB   71
         PRINT *, 'ENTER ONE LINE DESCRIPTION OF MATERIAL'               SMATDB   72
         READ *, MATDESC (INDEX)                                         SMATDB   73
         PRINT *, 'ENTER 7 ATTENUATION VALUES FROM LOW TO HIGH FREQ'     SMATDB   74
         READ *, (MATTEN(INDEX,C), C=1,7)                                SMATDB   75
         PRINT *, 'ENTER ATTENUATION QUALITY PERCENT'                    SMATDB   76
         READ *, QA (INDEX)                                              SMATDB   77
         PRINT *, 'ENTER 7 REFLECTION COEFFS FROM LOW TO HIGH FREQ'      SMATDB   78
         READ *, (MRCOEF(INDEX,C), C=1,7)                                SMATDB   79
         PRINT *, 'ENTER REFLECTION COEFICIENT QUALITY PERCENT'          SMATDB   80
         READ *, QR (INDEX)                                              SMATDB   81
         PRINT *                                                         SMATDB   82
         ENDIF                                                           SMATDB   83
      RETURN                                                             SMATDB   84
      END                                                                SMATDB   85
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 C          357B                             INTEGER
 ENTERED    362B                             LOGICAL
 INDEX      360B                             INTEGER
 MAT          0B  /MATC/                     CHAR*3         100
 MATDESC     36B  /MATC/                     CHAR*70        100
```

```
 MATID     361B                                 CHAR*3
 MATTEN      0B  /MATN/                         REAL           700
 MERR     4374B  /MATN/                         INTEGER
 MFREQ    3100B  /MATN/                         REAL           700
 MRCOEF   1274B  /MATN/                         REAL           700
 MTOT     4375B  /MATN/                         INTEGER
 QA       2570B  /MATN/                         REAL           100
 QR       2734B  /MATN/                         REAL           100
 R         356B                                 INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 MMAX    INTEGER                        100


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 VAL       INTEGER        1      FUNCTION


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10      5B                     20
    20  INACTIVE   DO-TERM         33


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 NEXT        4B     0


--STATISTICS--

 PROGRAM-UNIT LENGTH            366B =  246
 CM LABELLED COMMON LENGTH     5730B =  3032
 CM STORAGE USED              61000B =  25088
 COMPILE TIME                 0.117 SECONDS
```

```
                                                                        SMATDB    86
                                                                        SMATDB    87
      SUBROUTINE DISPLAY                                                SMATDB    88
************************************************************************COMM       1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                     ***COMM       2
************************************************************************COMM       3
      INTEGER MMAX                                                      COMM       4
      PARAMETER (MMAX=100)                                              COMM       5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM       6
     $              MFREQ(MMAX,7), MERR, MTOT                           COMM       7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                              COMM       8
      INTEGER MTOT, MERR                                                COMM       9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                COMM      10
      CHARACTER * 3  MAT                                                COMM      11
      CHARACTER * 70 MATDESC                                            COMM      12
************************************************************************COMM      13
************************************************************************COMM      14
      INTEGER R,C, COMMAND                                              SMATDB    90
      CHARACTER * 3 MATID                                               SMATDB    91
      LOGICAL FOUND                                                     SMATDB    92
      PRINT *, '(1) ALL MATERIALS  OR  (2) ONE MATERIAL'                SMATDB    93
      READ *, COMMAND                                                   SMATDB    94
      PRINT *                                                           SMATDB    95
      IF (COMMAND .EQ. 1) THEN                                          SMATDB    96
         PRINT *                                                        SMATDB    97
         DO 10 R = 1,MMAX                                               SMATDB    98
            MATID = MAT(R)                                              SMATDB    99
            IF  (MATID(1:1).EQ.'M' ) THEN                               SMATDB   100
                PRINT * , MAT(R)                                        SMATDB   101
                PRINT * , MATDESC (R)                                   SMATDB   102
                PRINT *, 'FREQUENCY: ' ,(MFREQ(R,C),C=1,7)              SMATDB   103
                PRINT * , 'ATTENUATION: ' ,(MATTEN(R,C), C=1,7)         SMATDB   104
                PRINT *,'ATTENUATION QUALITY PERCENT: ',QA(R)           SMATDB   105
                PRINT * , 'REFLECTION: ' , (MRCOEF(R,C), C=1,7)         SMATDB   106
                PRINT *,'REFLECTION COEF QUALITY PERCENT: ',QR(R)       SMATDB   107
                PRINT *                                                 SMATDB   108
                ENDIF                                                   SMATDB   109
 10      CONTINUE                                                       SMATDB   110
      ELSEIF (COMMAND .EQ. 2 ) THEN                                     SMATDB   111
         PRINT *, 'SPECIFY ID OF MATERIAL TO BE PRINTED (E.G. M05)'     SMATDB   112
         READ *, MATID                                                  SMATDB   113
         FOUND = .FALSE.                                                SMATDB   114
         DO 20 R = 1,MMAX                                               SMATDB   115
            IF (MAT(R) .EQ. MATID) THEN                                 SMATDB   116
               PRINT *, MAT(R)                                          SMATDB   117
               PRINT *, MATDESC(R)                                      SMATDB   118
               PRINT *, 'FREQUENCY: ' ,(MFREQ(R,C),C=1,7)               SMATDB   119
               PRINT *, 'ATTENUATION: ',(MATTEN(R,C), C=1,7)            SMATDB   120
               PRINT *, 'ATTENUATION QUALITY PERCENT: ',QA(R)           SMATDB   121
               PRINT *, 'REFLECTION: ',(MRCOEF(R,C), C=1,7)             SMATDB   122
               PRINT *, 'REFLECTION QUALITY PERCENT: ',QR(R)            SMATDB   123
               FOUND = .TRUE.                                           SMATDB   124
               ENDIF                                                    SMATDB   125
 20      CONTINUE                                                       SMATDB   126
         ENDIF                                                          SMATDB   127
            IF (.NOT. FOUND) THEN                                       SMATDB   128
               PRINT *, 'MATERIAL ', MATID, ' NOT FOUND'                SMATDB   129
               ENDIF                                                    SMATDB   130
         PRINT *                                                        SMATDB   131
      RETURN                                                            SMATDB   132
      END                                                               SMATDB   133
```

```
SUBROUTINE DISPLAY  74/175  OPT=0

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 C          511B                            INTEGER
 COMMAND    512B                            INTEGER
 FOUND      514B                            LOGICAL
 MAT          0B  /MATC/                    CHAR*3        100
 MATDESC     36B  /MATC/                    CHAR*70       100
 MATID      513B                            CHAR*3
 MATTEN       0B  /MATN/                    REAL          700
 MERR      4374B  /MATN/                    INTEGER
 MFREQ     3100B  /MATN/                    REAL          700
 MRCOEF    1274B  /MATN/                    REAL          700
 MTOT      4375B  /MATN/                    INTEGER
 QA        2570B  /MATN/                    REAL          100
 QR        2734B  /MATN/                    REAL          100
 R          510B                            INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE-----------------------VALUE

 MMAX    INTEGER                      100


--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM       38
    20  INACTIVE   DO-TERM       54


--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 DISPLAY     4B     0


--STATISTICS--

 PROGRAM-UNIT LENGTH             525B =  341
 CM LABELLED COMMON LENGTH      5730B =  3032
 CM STORAGE USED               61000B =  25088
 COMPILE TIME                 0.172 SECONDS
```

FUNCTION VAL 74/175 OPT=0

```
                                                             SMATDB   134
                                                             SMATDB   135
      INTEGER FUNCTION VAL(STRING)                           SMATDB   136
C** RETURNS THE INTEGER VALUE OF A STRING.                   SMATDB   137
      INTEGER NUMBER, X,L,EXP,DIGIT,GETLEN                   SMATDB   138
      CHARACTER * (*) STRING                                 SMATDB   139
      L = GETLEN(STRING)                                     SMATDB   140
      NUMBER = 0                                             SMATDB   141
      DO 10 X = L,1,-1                                       SMATDB   142
      EXP = L - X                                            SMATDB   143
      DIGIT = ICHAR(STRING(X:X)) - 16                        SMATDB   144
      NUMBER = NUMBER + DIGIT*10**EXP                        SMATDB   145
   10 CONTINUE                                               SMATDB   146
      VAL = NUMBER                                           SMATDB   147
      RETURN                                                 SMATDB   148
      END                                                    SMATDB   149
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
DIGIT	76B			INTEGER	
EXP	75B			INTEGER	
L	74B			INTEGER	
NUMBER	72B			INTEGER	
STRING	1		DUMMY-ARG	CHAR*(*)	
VAL	71B			INTEGER	
X	73B			INTEGER	

--PROCEDURES--(LO=A)

-NAME------	TYPE--------	ARGS------	CLASS-----
GETLEN	INTEGER	1	FUNCTION
ICHAR	INTEGER	1	INTRINSIC

--STATEMENT LABELS--(LO=A)

-LABEL-	ADDRESS-----	PROPERTIES----	DEF
10	INACTIVE	DO-TERM	13

--ENTRY POINTS--(LO=A)

-NAME---	ADDRESS--	ARGS---
VAL	6B	1

--STATISTICS--

PROGRAM-UNIT LENGTH	102B = 66
CM STORAGE USED	61000B = 25088
COMPILE TIME	0.041 SECONDS

```


      INTEGER FUNCTION GETLEN (STRING)
C
C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING
C
C
C ARGUMENT DEFINITIONS --
C   INPUT ARGUMENTS
C     STRING - STRING WHOSE LENGTH IS TO BE DETERMINED
C
      CHARACTER * (*) STRING
C
C FUNCTION PARAMETERS
      CHARACTER * 1 BLANK
      PARAMETER (BLANK = ' ')
C
C LOCAL VARIABLES
      INTEGER NEXT
C
C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK
      DO 10 NEXT = LEN(STRING),1,-1
         IF (STRING(NEXT : NEXT) .NE. BLANK) THEN
            GETLEN = NEXT
            RETURN
         ENDIF
   10 CONTINUE
C
C ALL CHARACTERS ARE BLANKS
      GETLEN = 0
C
      RETURN
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  GETLEN     63B                             INTEGER
  NEXT       64B                             INTEGER
  STRING      1   DUMMY-ARG                  CHAR*(*)


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-----------------------VALUE

  BLANK   CHAR*1                        ' '


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  LEN       INTEGER       1       INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        27
```



```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  GETLEN      6B      1


--STATISTICS--

  PROGRAM-UNIT LENGTH                 70B =  56
  CM STORAGE USED                  61000B =  25088
  COMPILE TIME                     0.038 SECONDS
```

```
                                                                            SMATDB     183
                                                                            SMATDB     184
      SUBROUTINE QUIT                                                       SMATDB     185
***********************************************************************COMM          1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                        ***COMM       2
***********************************************************************COMM          3
      INTEGER MMAX                                                          COMM       4
      PARAMETER (MMAX=100)                                                  COMM       5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX),     COMM       6
     $              MFREQ(MMAX,7), MERR, MTOT                               COMM       7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                                  COMM       8
      INTEGER MTOT, MERR                                                    COMM       9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                    COMM      10
      CHARACTER * 3  MAT                                                    COMM      11
      CHARACTER * 70 MATDESC                                                COMM      12
***********************************************************************COMM         13
***********************************************************************COMM         14
      INTEGER R,C                                                           SMATDB     187
      CHARACTER * 3 MATID                                                   SMATDB     188
      LOGICAL FOUND                                                         SMATDB     189
      OPEN (UNIT =6,FILE='MATTER',FORM='FORMATTED',                         SMATDB     190
     Z     ACCESS='SEQUENTIAL', STATUS='NEW')                               SMATDB     191
      REWIND (6)                                                            SMATDB     192
      DO 10 R=1,MMAX                                                        SMATDB     193
         MATID = MAT(R)                                                     SMATDB     194
         IF (MATID(1:1) .EQ. 'M' ) THEN                                     SMATDB     195
            WRITE (6,100) MAT(R)                                            SMATDB     196
            WRITE (6,200) MATDESC (R)                                       SMATDB     197
            WRITE (6,400) 1E+4, 1E+5 ,1E+6, 1E+7, 1E+8, 1E+9, 1E+10         SMATDB     198
            WRITE (6,400) (MATTEN(R,C),C=1,7)                               SMATDB     199
            WRITE (6,400) QA(R)                                             SMATDB     200
            WRITE (6,400) (MRCOEF(R,C),C=1,7)                               SMATDB     201
            WRITE (6,400) QR(R)                                             SMATDB     202
            ENDIF                                                           SMATDB     203
10    CONTINUE                                                              SMATDB     204
      ENDFILE (6)                                                           SMATDB     205
      CALL PF ('REPLACE',0,'MATTER')                                        SMATDB     206
      CLOSE (6, STATUS = 'DELETE')                                          SMATDB     207
100   FORMAT (A3)                                                           SMATDB     208
200   FORMAT (A70)                                                          SMATDB     209
400   FORMAT (7(1X, E9.3))                                                  SMATDB     210
      RETURN                                                                SMATDB     211
      END                                                                   SMATDB     212
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
C	270B			INTEGER	
FOUND	NONE		UNUSED/*S*	LOGICAL	
MAT	0B	/MATC/		CHAR*3	100
MATDESC	36B	/MATC/		CHAR*70	100
MATID	271B			CHAR*3	
MATTEN	0B	/MATN/		REAL	700
MERR	4374B	/MATN/		INTEGER	
MFREQ	3100B	/MATN/		REAL	700
MRCOEF	1274B	/MATN/		REAL	700
MTOT	4375B	/MATN/		INTEGER	
QA	2570B	/MATN/		REAL	100
QR	2734B	/MATN/		REAL	100
R	267B			INTEGER	

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  MMAX     INTEGER                        100


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  PF                        3      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        35
    100     164B    FORMAT         39
    200     166B    FORMAT         40
    400     170B    FORMAT         41


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  QUIT        5B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE6   AUX/FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH             277B =  191
  CM LABELLED COMMON LENGTH      5730B =  3032
  CM STORAGE USED               61000B =  25088
  COMPILE TIME                  0.101 SECONDS
```

```
                                                                        SMATDB     213
                                                                        SMATDB     214
                                                                        SMATDB     215
                                                                        SMATDB     216
      SUBROUTINE CHANGE                                                 SMATDB     217
************************************************************************COMM         1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                     ***COMM         2
************************************************************************COMM         3
      INTEGER MMAX                                                      COMM         4
      PARAMETER (MMAX=100)                                              COMM         5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM         6
     $               MFREQ(MMAX,7), MERR, MTOT                          COMM         7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                              COMM         8
      INTEGER MTOT, MERR                                                COMM         9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                COMM        10
      CHARACTER * 3  MAT                                                COMM        11
      CHARACTER * 70 MATDESC                                            COMM        12
************************************************************************COMM        13
************************************************************************COMM        14
      INTEGER R,C,COMMAND, VAL, INDEX                                   SMATDB     219
      CHARACTER * 3 MATID                                               SMATDB     220
      LOGICAL FOUND                                                     SMATDB     221
      PRINT *, 'ENTER MATERIAL I.D. OF GROUP TO BE CHANGED'             SMATDB     222
         READ *, MATID                                                  SMATDB     223
         FOUND = .FALSE.                                                SMATDB     224
         DO 20 R = 1,MMAX                                               SMATDB     225
            IF (MAT(R) .EQ. MATID) THEN                                 SMATDB     226
               PRINT *                                                  SMATDB     227
               PRINT *, 'LINE 1:    MATERIAL ID '                       SMATDB     228
               PRINT *, MAT (R)                                         SMATDB     229
               PRINT *, 'LINE 2:     DESCRIPTION'                       SMATDB     230
               PRINT *, MATDESC (R)                                     SMATDB     231
               PRINT *, 'LINE 3:      FREQUENCIES'                      SMATDB     232
               PRINT *, (MFREQ (R,C), C=1,7)                            SMATDB     233
               PRINT *, 'LINE 4:      ATTENUATIONS'                     SMATDB     234
               PRINT *, (MATTEN (R,C), C=1,7)                           SMATDB     235
               PRINT *, 'LINE 5:      ATTENUATION QUALITY PERCENT'      SMATDB     236
               PRINT *, QA (R)                                          SMATDB     237
               PRINT *, 'LINE 6:     REFLECTION COEFFICIENTS'           SMATDB     238
               PRINT *, (MRCOEF (R,C), C=1,7)                           SMATDB     239
               PRINT *, 'LINE 7:  REFLECTION QUALITY PERCENT'           SMATDB     240
               PRINT *, QR (R)                                          SMATDB     241
               FOUND = .TRUE.                                           SMATDB     242
               INDEX = R                                                SMATDB     243
               ENDIF                                                    SMATDB     244
 20      CONTINUE                                                       SMATDB     245
            IF (.NOT. FOUND) THEN                                       SMATDB     246
               PRINT *, 'MATERIAL ', MATID, ' NOT FOUND'                SMATDB     247
               RETURN                                                   SMATDB     248
               ENDIF                                                    SMATDB     249
 30   PRINT *                                                           SMATDB     250
      PRINT *, 'ENTER NUMBER OF LINE TO BE CHANGED',                    SMATDB     251
     Z    '       (99 TO END CHANGES)'                                  SMATDB     252
      READ *, COMMAND                                                   SMATDB     253
      IF (COMMAND .EQ. 1) THEN                                          SMATDB     254
         PRINT *, 'ENTER NEW I.D.'                                      SMATDB     255
         MAT (INDEX) = '   '                                            SMATDB     256
         READ *, MATID                                                  SMATDB     257
         R = VAL ( MATID(2:3) )                                         SMATDB     258
         MAT (R) = MATID                                                SMATDB     259
         MATDESC (R) = MATDESC (INDEX)                                  SMATDB     260
         MFREQ (R,1) = MFREQ (INDEX,1)                                  SMATDB     261
         MFREQ (R,2) = MFREQ (INDEX,2)                                  SMATDB     262
         MFREQ (R,3) = MFREQ (INDEX,3)                                  SMATDB     263
```

```
            MFREQ (R,4) = MFREQ (INDEX,4)
            MFREQ (R,5) = MFREQ (INDEX,5)
            MFREQ (R,6) = MFREQ (INDEX,6)
            MFREQ (R,7) = MFREQ (INDEX,7)
            MATTEN (R,1) = MATTEN (INDEX,1)
            MATTEN (R,2) = MATTEN (INDEX,2)
            MATTEN (R,3) = MATTEN (INDEX,3)
            MATTEN (R,4) = MATTEN (INDEX,4)
            MATTEN (R,5) = MATTEN (INDEX,5)
            MATTEN (R,6) = MATTEN (INDEX,6)
            MATTEN (R,7) = MATTEN (INDEX,7)
            QA (R) = QA (INDEX)
            MRCOEF (R,1) = MRCOEF (INDEX,1)
            MRCOEF (R,2) = MRCOEF (INDEX,2)
            MRCOEF (R,3) = MRCOEF (INDEX,3)
            MRCOEF (R,4) = MRCOEF (INDEX,4)
            MRCOEF (R,5) = MRCOEF (INDEX,5)
            MRCOEF (R,6) = MRCOEF (INDEX,6)
            MRCOEF (R,7) = MRCOEF (INDEX,7)
            QR (R) = QR (INDEX)
            INDEX = R
         ELSE IF ( COMMAND .EQ. 2 ) THEN
            PRINT *, 'ENTER NEW ONE LINE DESCRIPTION OF MATERIAL'
            READ *, MATDESC (INDEX)
         ELSE IF ( COMMAND .EQ. 3 ) THEN
            PRINT *, 'ENTER NEW SET OF 7 FREQUENCIES'
            READ *, (MFREQ (INDEX,C), C=1,7)
         ELSE IF ( COMMAND .EQ. 4 ) THEN
            PRINT *, 'ENTER NEW SET OF 7 ATTENUATIONS'
            READ *, (MATTEN (INDEX,C), C=1,7)
         ELSE IF ( COMMAND .EQ. 5 ) THEN
            PRINT *, 'ENTER NEW ATTENUATION QUALITY PERCENT'
            READ *, QA (INDEX)
         ELSE IF ( COMMAND .EQ. 6 ) THEN
            PRINT *, 'ENTER NEW SET OF 7 REFLECTION COEFFICIENTS'
            READ *, (MRCOEF (INDEX,C), C=1,7)
         ELSE IF ( COMMAND .EQ. 7 ) THEN
            PRINT *, 'ENTER NEW REFLECTION QUALITY PERCENT'
            READ *, QR (INDEX)
         ELSE IF ( COMMAND .EQ. 99 ) THEN
            GOTO 40
            ENDIF
         GOTO 30
C
   40    PRINT *
         PRINT *, MAT (INDEX)
         PRINT *, MATDESC (INDEX)
         PRINT *, (MFREQ (INDEX,C), C=1,7)
         PRINT *, (MATTEN (INDEX,C), C=1,7)
         PRINT *, QA (INDEX)
         PRINT *, (MRCOEF (INDEX,C), C=1,7)
         PRINT *, QR (INDEX)
         PRINT *
         RETURN
         END
```

```
SUBROUTINE CHANGE    74/175  OPT=0

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  C          1137B                             INTEGER
  COMMAND    1140B                             INTEGER
  FOUND      1143B                             LOGICAL
  INDEX      1141B                             INTEGER
  MAT           0B  /MATC/                     CHAR*3          100
  MATDESC      36B  /MATC/                     CHAR*70         100
  MATID      1142B                             CHAR*3
  MATTEN        0B  /MATN/                     REAL            700
  MERR       4374B  /MATN/                     INTEGER
  MFREQ      3100B  /MATN/                     REAL            700
  MRCOEF     1274B  /MATN/                     REAL            700
  MTOT       4375B  /MATN/                     INTEGER
  QA         2570B  /MATN/                     REAL            100
  QR         2734B  /MATN/                     REAL            100
  R          1136B                             INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  MMAX    INTEGER                        100


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  VAL       INTEGER        1      FUNCTION


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     20  INACTIVE   DO-TERM        46
     30    204B                    51
     40    477B                   109


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  CHANGE      5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH          1160B =   624
  CM LABELLED COMMON LENGTH    5730B =  3032
  CM STORAGE USED             63000B = 26112
  COMPILE TIME                0.318 SECONDS
```

SUBROUTINE LMATTER 74/175 OPT=0

```
      SUBROUTINE LMATTER                                                LMATTER     1
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LMATTER     2
*!!!                                                                 !!!LMATTER     3
*!!!   THIS SUBROUTINE LOADS THE MATERIAL DATABASE INTO ARRAYS FOR   !!!LMATTER     4
*!!!   FURTHER PROGRAM USE.                                          !!!LMATTER     5
*!!!                                                                 !!!LMATTER     6
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LMATTER     7
************************************************************************LMATTER     8
************************************************************************COMM        1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                     ***COMM        2
************************************************************************COMM        3
      INTEGER MMAX                                                      COMM        4
      PARAMETER (MMAX=100)                                              COMM        5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM        6
     $              MFREQ(MMAX,7), MERR, MTOT                           COMM        7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                              COMM        8
      INTEGER MTOT, MERR                                                COMM        9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                COMM       10
      CHARACTER * 3  MAT                                                COMM       11
      CHARACTER * 70 MATDESC                                            COMM       12
************************************************************************COMM       13
************************************************************************COMM       14
**************************************************                      LMATTER    10
*   DECLARATION OF VARIABLES                                            LMATTER    11
**************************************************                      LMATTER    12
      INTEGER R, C, VAL                                                 LMATTER    13
      CHARACTER * 3 MATID                                               LMATTER    14
**************************************************                      LMATTER    15
*  GET FILE                                                             LMATTER    16
**************************************************                      LMATTER    17
      MERR = 0                                                          LMATTER    18
      CALL PF ('GET',0,'MATTER','RC',MERR)                              LMATTER    19
**************************************************                      LMATTER    20
*  FILE ERROR CHECK                                                     LMATTER    21
**************************************************                      LMATTER    22
      IF ( MERR .EQ. 0 ) THEN                                           LMATTER    23
999     CONTINUE                                                        LMATTER    24
      ELSE IF ( MERR .EQ. 2 ) THEN                                      LMATTER    25
        CALL WARNING (3)                                                LMATTER    26
        RETURN                                                          LMATTER    27
      ELSE                                                              LMATTER    28
        CALL WARNING (4)                                                LMATTER    29
        RETURN                                                          LMATTER    30
      END IF                                                            LMATTER    31
**************************************************                      LMATTER    32
*   OPEN FILE                                                           LMATTER    33
**************************************************                      LMATTER    34
      OPEN (UNIT = 3, FILE='MATTER',FORM='FORMATTED',                   LMATTER    35
     $      STATUS = 'OLD', ACCESS = 'SEQUENTIAL')                      LMATTER    36
      REWIND (3)                                                        LMATTER    37
**************************************************                      LMATTER    38
*  INITIALIZE ARRAYS                                                    LMATTER    39
**************************************************                      LMATTER    40
      DATA MAT / 100 * '   ' /                                          LMATTER    41
      DATA MATDESC / 100 * '   ' /                                      LMATTER    42
      DATA MFREQ / 700 * 0.0 /                                          LMATTER    43
      DATA MATTEN / 700 * 0.0 /                                         LMATTER    44
      DATA QA / 100 * 0.0 /                                             LMATTER    45
      DATA MRCOEF / 700 * 0.0 /                                         LMATTER    46
      DATA QR / 100 * 0.0 /                                             LMATTER    47
**************************************************                      LMATTER    48
*  READ IN THE MATERIAL FILE                                            LMATTER    49
**************************************************                      LMATTER    50
10    READ (3,1000,END=20) MATID                                        LMATTER    51
```

```
      R = VAL(MATID(2:3))
      MAT (R) = MATID
      READ (3,2000,END=20) MATDESC (R)
      READ (3,4000,END=20) (MFREQ(R,C),C=1,7)
      READ (3,4000,END=20) (MATTEN(R,C),C=1,7)
      READ (3,4000,END=20) QA (R)
      READ (3,4000,END=20) (MRCOEF(R,C),C=1,7)
      READ (3,4000,END=20) QR (R)
      GOTO 10
20    CONTINUE
1000  FORMAT (A3)
2000  FORMAT (A70)
4000  FORMAT (7(1X,E9.3))
*************************************************
*  CLOSE FILE
*************************************************
      CLOSE (3,STATUS = 'DELETE')
      RETURN
      END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	--BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
C	327B			INTEGER	
MAT	0B	/MATC/		CHAR*3	100
MATDESC	36B	/MATC/		CHAR*70	100
MATID	330B			CHAR*3	
MATTEN	0B	/MATN/		REAL	700
MERR	4374B	/MATN/		INTEGER	
MFREQ	3100B	/MATN/		REAL	700
MRCOEF	1274B	/MATN/		REAL	700
MTOT	4375B	/MATN/		INTEGER	
QA	2570B	/MATN/		REAL	100
QR	2734B	/MATN/		REAL	100
R	326B			INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME----	TYPE------------------------	VALUE
MMAX	INTEGER	100

--PROCEDURES--(LO=A)

-NAME------	TYPE--------	ARGS------	CLASS-----
PF		5	SUBROUTINE
VAL	INTEGER	1	FUNCTION
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-	ADDRESS-----	PROPERTIES----	DEF	-LABEL-	ADDRESS-----	PROPERTIES----	DEF
10	36B		64	1000	202B	FORMAT	75
20	160B		74	2000	204B	FORMAT	76
999	*NO REFS*		37	4000	206B	FORMAT	77

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  LMATTER     5B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE3   AUX/FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH              336B =  222
  CM LABELLED COMMON LENGTH       5730B =  3032
  CM STORAGE USED                61000B =  25088
  COMPILE TIME                   0.127 SECONDS
```

```
      1        SUBROUTINE WARNING(ERR)                                          WARNING      1
      2        INTEGER ERR, ERRM                                                WARNING      2
      3        CHARACTER*45 MESSAGE(20)                                         WARNING      3
      4        DATA MESSAGE( 1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING      4
      5        DATA MESSAGE( 2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE    '/ WARNING      5
      6        DATA MESSAGE( 3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG   '/ WARNING      6
      7        DATA MESSAGE( 4)/'FILE HANDLING PROBLEM ON "MATTER FILE        '/ WARNING      7
      8        DATA MESSAGE( 5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING      8
      9        DATA MESSAGE( 6)/'FILE HANDLING PROBLEM ON "TYPE" FILE         '/ WARNING      9
     10        DATA MESSAGE( 7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING     10
     11        DATA MESSAGE( 8)/'FILE HANDLING PROBLEM ON "WALL" FILE         '/ WARNING     11
     12        DATA MESSAGE( 9)/'HEIGHT AND WIDTH OF ROOM MISSING             '/ WARNING     12
     13        DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING                    '/ WARNING     13
     14        DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG       '/ WARNING     14
     15        DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE         '/ WARNING     15
     16        DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING     16
     17        DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING     17
     18        DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING     18
     19        DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING     19
     20        DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING     20
     21        DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING     21
     22        DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING     22
     23        DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING     23
     24        ERRM=12                                                          WARNING     24
     25        IERR = ERR                                                       WARNING     25
     26        IF(ERR.GT.ERRM) IERR=20                                          WARNING     26
     27        WRITE(6,20)                                                      WARNING     27
     28        WRITE(6,10) ERR,MESSAGE(IERR)                                    WARNING     28
     29        WRITE(6,20)                                                      WARNING     29
     30 10     FORMAT(' ***WARNING NUMBER = ',I5,' ***  ',A45)                  WARNING     30
     31 20     FORMAT('     ')                                                  WARNING     31
     32         RETURN                                                          WARNING     32
     33         END                                                             WARNING     33
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 ERR          1   DUMMY-ARG                  INTEGER
 ERRM       60B                              INTEGER
 IERR      213B                              INTEGER
 MESSAGE    61B                              CHAR*45          20


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10      34B     FORMAT          30
    20      42B     FORMAT          31


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 WARNING      5B      1
```

```
FTN 5.1+552          83/12/24. 09.48.06 PAGE     19
SUBROUTINE WARNING  74/175  OPT=0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

 TAPE6   FMT/SEQ


--STATISTICS--

 PROGRAM-UNIT LENGTH              216B =   142
 CM STORAGE USED                61000B =  25088
 COMPILE TIME                   0.063 SECONDS
```

```
      SUBROUTINE ERROR(IERR)                                            ERROR      1
      CHARACTER*45 MESSAGE(20)                                          ERROR      2
      DATA MESSAGE( 1)/'MATERIALS DATA BASE IS EMPTY                 '/ ERROR      3
      DATA MESSAGE( 2)/'FREQUENCY IS OUT OF RANGE                    '/ ERROR      4
      DATA MESSAGE( 3)/'THIS MATERIAL IS NOT IN DATA BASE            '/ ERROR      5
      DATA MESSAGE( 4)/'DENOMINATOR IS ZERO                          '/ ERROR      6
      DATA MESSAGE( 5)/'FILE HANDLING ERROR                          '/ ERROR      7
      DATA MESSAGE( 6)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR      8
      DATA MESSAGE( 7)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR      9
      DATA MESSAGE( 8)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     10
      DATA MESSAGE( 9)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     11
      DATA MESSAGE(10)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     12
      DATA MESSAGE(11)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     13
      DATA MESSAGE(12)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     14
      DATA MESSAGE(13)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     15
      DATA MESSAGE(14)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     16
      DATA MESSAGE(15)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     17
      DATA MESSAGE(16)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     18
      DATA MESSAGE(17)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     19
      DATA MESSAGE(18)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     20
      DATA MESSAGE(19)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     21
      DATA MESSAGE(20)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     22
      IERRM=5                                                           ERROR     23
      IF(IERR.GT.IERRM) IERR=20                                         ERROR     24
      WRITE(6,10) IERR,MESSAGE(IERR)                                    ERROR     25
   10 FORMAT(' ***ERROR NUMBER = ',I5,' ***  ',A45)                     ERROR     26
      CALL PMDSTOP                                                      ERROR     27
      STOP 'ERROR'                                                      ERROR     28
      END                                                               ERROR     29
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  IERR        1   DUMMY-ARG                  INTEGER
  IERRM     210B                             INTEGER
  MESSAGE    56B                             CHAR*45        20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  PMDSTOP                   0      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10     36B     FORMAT          26


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ERROR      5B     1
```



```
--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE6   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH           213B =  139
  CM STORAGE USED             61000B =  25088
  COMPILE TIME                 0.056 SECONDS
```

Appendix 9.3 Listing of Computer Program SHOLES

```
      PROGRAM SHOLES (INPUT,TAPE1=INPUT)                                SHOLES   1
*                                                                       SHOLES   2
*THIS INTERACTIVE PROGRAM INPUTS THE DATA DESCRIBING EACH HOLE          SHOLES   3
*IN THE BUILDING AND STORES IT.  THE FILE NAME IS CREATED BY            SHOLES   4
*ATTACHING "B" TO THE FRONT OF AND "H" TO THE BACK OF THE BUILDING      SHOLES   5
*IDENTIFICATION. THE BUILDING IDENTIFICATION CAN BE NO MORE             SHOLES   6
*THAN 5 ALPHANUMERIC CHARACTERS.                                        SHOLES   7
                                                                        SHOLES   8
************************************************************************COMF     1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF     2
************************************************************************COMF     3
      INTEGER FMAX                                                      COMF     4
      PARAMETER (FMAX = 50)                                             COMF     5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF     6
     $                 FTOT                                             COMF     7
      COMMON /INITILC/ BLDG                                             COMF     8
      CHARACTER * 5 BLDG                                                COMF     9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF    10
      INTEGER QUALITY, FERR, FTOT                                       COMF    11
************************************************************************COMF    12
************************************************************************COMF    13
************************************************************************COMR     1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR     2
************************************************************************COMR     3
      INTEGER RMAX                                                      COMR     4
      PARAMETER (RMAX = 20)                                             COMR     5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR     6
      INTEGER NROOMS                                                    COMR     7
      REAL ROOM                                                         COMR     8
************************************************************************COMR     9
************************************************************************COMR    10
************************************************************************COMH     1
***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH     2
************************************************************************COMH     3
      INTEGER HMAX                                                      COMH     4
      PARAMETER (HMAX = 35)                                             COMH     5
      COMMON /HOLEN/ HTOT, HERR                                         COMH     6
      COMMON /HOLEC/ HOLE(HMAX,4)                                       COMH     7
      INTEGER HTOT, HERR                                                COMH     8
      CHARACTER * 3 HOLE                                                COMH     9
* ==================================================                    COMH    10
* DESCRIPTION OF ARRAYS                                                 COMH    11
* ==================================================                    COMH    12
*     ROOM IDENTIFICATION              APERTURE ID                      COMH    13
* ----------------------------------                                    COMH    14
* DIRECTION    FROM ROOM    TO ROOM                                     COMH    15
* --------------------------------------------------                    COMH    16
* HOLE(X,1)    HOLE(X,2)    HOLE(X,3)    HOLE(X,4)                      COMH    17
*   A3            A3           A3           A3                          COMH    18
************************************************************************COMH    19
************************************************************************COMH    20
      INTEGER GETLEN,QUIT,ABORT,ANSWER,OLDFILE,N,Y1,Y2,LINE             SHOLES  12
      INTEGER IERR                                                      SHOLES  13
      CHARACTER * 7  PFN                                                SHOLES  14
*                                                                       SHOLES  15
* INITIALIZATION                                                        SHOLES  16
      QUIT = 0                                                          SHOLES  17
      HTOT = 0                                                          SHOLES  18
      ABORT = 0                                                         SHOLES  19
100   PRINT*                                                            SHOLES  20
      PRINT *, 'ENTER BUILDING IDENTIFICATION  (E.G. ''101'')'          SHOLES  21
      PRINT *, '    (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)'           SHOLES  22
      REWIND 1                                                          SHOLES  23
      READ(1,*,END=100) BLDG                                            SHOLES  24
```

```

      IF ( GETLEN(BLDG) .GT. 5 ) THEN
        GO TO 100
      END IF
      PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'H'
*
*** LOAD DATA FROM EXISTING FILE IF NECESSARY
200   PRINT*
      PRINT*,'WILL THIS BE'
      PRINT*,'    (1) A MODIFICATION OF AN EXISTING FILE?'
      PRINT*,'    (2) A NEW FILE?'
      PRINT*,'ENTER A NUMBER  !!!'
      REWIND 1
      READ(1,*,END=200) OLDFILE
      IF ( ( OLDFILE .NE. 1 ) .AND. ( OLDFILE .NE. 2 ) ) THEN
        GOTO 200
      ELSE IF ( OLDFILE .EQ. 1 ) THEN
*
*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME
        IERR = 0
        CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)
        IF ( IERR .EQ. 2 ) THEN
          PRINT*
          PRINT *, 'FILE ',PFN, ' NOT FOUND'
          PRINT*, 'PROGRAM ABORTED!!!'
          PRINT*
          PRINT*, 'FIND CORRECT BUILDING IDENTIFIER AND RESTART ',
     +              'PROGRAM'
          PRINT*
          STOP
*
        ELSE
          CALL LHOLE
          IF (HERR .NE. 0) CALL ERROR(5)
        END IF
      ELSE IF ( OLDFILE .EQ. 2 ) THEN
*
*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME
        IERR = 0
        CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)
        IF ( IERR .EQ. 0 ) THEN
          PRINT*
          PRINT*, 'DATA FILE ALREADY EXISTS FOR BUILDING ',BLDG
          PRINT*
          PRINT*,'IF YOU ENTER DATA AND STORE IT, YOU WILL WRITE ',
     +             'OVER THE OLD FILE.'
250       PRINT*
          PRINT*,'YOU MAY EITHER (1) ABORT OR (2) CONTINUE.'
          PRINT*,'INDICATE YOUR CHOICE BY ENTERING A NUMBER.'
          REWIND 1
          READ(1,*,END=250) ANSWER
          IF ( ANSWER .EQ. 1 ) THEN
            PRINT*
            PRINT*,'PROGRAM HAS BEEN ABORTED, AT YOUR REQUEST'
            PRINT*
            STOP
          ELSE IF ( ANSWER .EQ. 2 ) THEN
            CONTINUE
TRIVIAL* CONTINUE WITH NO STATEMENT LABEL -- IGNORED
          ELSE
            GOTO 250
          END IF
        ELSE IF ( IERR .EQ. 2 ) THEN
*
```

```
*** NO DATA FILE ALREADY EXISTS FOR THIS BUILDING AND DATA ENTRY
*** CAN CONTINUE
            CONTINUE
TRIVIAL* CONTINUE WITH NO STATEMENT LABEL -- IGNORED
          ELSE
*
*** PERMANENT FILE ERROR
            PRINT*
            PRINT*,'PROGRAM ABORTED !!!'
            PRINT*,'   SOME PERMANENT FILE ERROR HAS OCCURRED.'
            PRINT*,'   DOUBLE CHECK YOUR BUILDING IDENTIFICATION ',
     +             'AND TRY AGAIN'
            STOP
          END IF
*
          PRINT*
          PRINT*, ' BEGIN ENTERING DATA'
 300      HTOT = HTOT + 1
          IF ( HTOT .EQ. 1) THEN
            CALL DATAIN(1,HTOT)
          ELSE
            CALL DATAIN (0,HTOT)
          END IF
 400      PRINT*
          PRINT*, 'DO YOU WANT TO ENTER MORE DATA?',
     +             '(1) YES  (2) NO'
          PRINT*, '  ENTER A NUMBER !!!'
          REWIND 1
          READ(1,*,END=400) ANSWER
          IF ( (ANSWER .NE. 1) .AND. (ANSWER .NE. 2) ) THEN
            GOTO 400
          ELSE IF ( ANSWER .EQ. 1) THEN
            GOTO 300
          ELSE IF ( ANSWER .EQ. 2 ) THEN
            PRINT*
            PRINT*, 'DATA ENTRY DISCONTINUED'
          END IF
        END IF
*
*** MANIPULATE DATA
        CALL MANIP (QUIT,ABORT)
*
*** TERMINATE PROGRAM, STORING DATA IF NECESSARY
        IF ( QUIT .EQ. 1 ) THEN
         OPEN(UNIT=6,FILE=PFN(1:GETLEN(PFN)),FORM='FORMATTED',
     +           ACCESS='SEQUENTIAL',STATUS='NEW')
 500      FORMAT (1X,4(1X,A3))
          DO 600 N = 1,HTOT
            WRITE (6,500)(HOLE(N,Y1), Y1=1,4)
 600      CONTINUE
          ENDFILE(6)
          CALL PF ('REPLACE',0,PFN(1:GETLEN(PFN)))
WARNING* NUMBER OF ARGUMENTS IN REFERENCE TO _PF IS NOT CONSISTENT
          CLOSE(6,STATUS='DELETE')
          PRINT*
          PRINT*,'DATA HAS BEEN STORED AND PROGRAM TERMINATED'
        END IF
        IF( ABORT .EQ. 1 ) THEN
          PRINT*
          PRINT*, 'PROGRAM HAS BEEN ABORTED'
          PRINT*,'     NO DATA HAS BEEN STORED !!!'
        END IF
        STOP
        END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ABORT    1054B                             INTEGER
  AFLAG       2B   /INITILN/                 REAL
  ANSWER   1055B                             INTEGER
  BLDG        0B   /INITILC/                 CHAR*5
  FERR       66B   /INITILN/                 INTEGER
  FREQ        0B   /INITILN/                 REAL
  FREGA       4B   /INITILN/                 REAL             50
  FTOT       67B   /INITILN/                 INTEGER
  HERR        1B   /HOLEN/                   INTEGER
  HOLE        0B   /HOLEC/                   CHAR*3          140
  HTOT        0B   /HOLEN/                   INTEGER
  IERR     1061B                             INTEGER
  LINE     NONE             UNUSED/*S*       INTEGER
  N        1057B                             INTEGER
  NROOMS   1244B   /ROOMN/                   INTEGER
  OLDFILE  1056B                             INTEGER
  PFN      1062B                             CHAR*7
  QUALITY     1B   /INITILN/                 INTEGER
  QUIT     1053B                             INTEGER
  RAREA    1245B   /ROOMN/                   REAL             20
  RFLAG       3B   /INITILN/                 REAL
  ROOM        0B   /ROOMN/                   REAL            676
  Y1       1060B                             INTEGER
  Y2       NONE             UNUSED/*S*       INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-----------------------VALUE

  FMAX     INTEGER                        50
  HMAX     INTEGER                        35
  RMAX     INTEGER                        20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  DATAIN                   2      SUBROUTINE       LHOLE                    0      SUBROUTINE
  ERROR                    1      SUBROUTINE       MANIP                    2      SUBROUTINE
  GETLEN    INTEGER        1      FUNCTION         PF                       5      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    100     21B                    60          400     256B                   150
    200     47B                    72          500     606B     FORMAT        173
    250    166B                   111          600  INACTIVE    DO-TERM       176
    300    244B                   144


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  SHOLES     14B     0
```

```
--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-------------

 TAPE1   FMT/SEQ
 TAPE6   AUX/FMT/SEQ


--STATISTICS--

 PROGRAM-UNIT LENGTH           1065B =  565
 CM LABELLED COMMON LENGTH     1436B =  798
 CM STORAGE USED              63000B = 26112
 COMPILE TIME                 0.270 SECONDS

      2  TRIVIAL   ERRORS IN SHOLES

      1  WARNING   ERROR IN  SHOLES
```

```
      SUBROUTINE DATAIN (INSERT,LINE)                                 SHOLES     150
************************************************************************COMR        1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR        2
************************************************************************COMR        3
      INTEGER RMAX                                                    COMR        4
      PARAMETER (RMAX = 20)                                           COMR        5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)    COMR        6
      INTEGER NROOMS                                                  COMR        7
      REAL ROOM                                                       COMR        8
************************************************************************COMR        9
************************************************************************COMR       10
************************************************************************COMH        1
***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH        2
************************************************************************COMH        3
      INTEGER HMAX                                                    COMH        4
      PARAMETER (HMAX = 35)                                           COMH        5
      COMMON /HOLEN/ HTOT, HERR                                       COMH        6
      COMMON /HOLEC/ HOLE(HMAX,4)                                     COMH        7
      INTEGER HTOT, HERR                                              COMH        8
      CHARACTER * 3 HOLE                                              COMH        9
* ==================================================                  COMH       10
* DESCRIPTION OF ARRAYS                                               COMH       11
* ==================================================                  COMH       12
*     ROOM IDENTIFICATION              APERTURE ID                    COMH       13
* -----------------------------------                                 COMH       14
* DIRECTION    FROM ROOM     TO ROOM                                  COMH       15
* --------------------------------------------------                  COMH       16
* HOLE(X,1)    HOLE(X,2)     HOLE(X,3)    HOLE(X,4)                   COMH       17
*   A3            A3            A3           A3                       COMH       18
************************************************************************COMH       19
************************************************************************COMH       20
      INTEGER  ANSWER,LOK,DOK,NOK,GETLEN,VAL, INSERT,LINE,V           SHOLES     153
      CHARACTER *3 DIR,FROM,TO,ID                                     SHOLES     154
200     PRINT*                                                        SHOLES     155
        PRINT*, 'ENTER DIRECTION (E. G. ''LR'')'                      SHOLES     156
        REWIND 1                                                      SHOLES     157
        READ(1,*,END=200) DIR                                         SHOLES     158
        IF ((DIR .NE. 'LR')                                           SHOLES     159
     +        .AND. (DIR .NE. 'FB')                                   SHOLES     160
     +        .AND. (DIR .NE. 'UD')) THEN                             SHOLES     161
          PRINT*, 'DIRECTION MUST BE ''LR'' OR ''FB'' OR ''UD'''      SHOLES     162
          PRINT*, 'TRY AGAIN!!!'                                      SHOLES     163
          GOTO 200                                                    SHOLES     164
        END IF                                                        SHOLES     165
        HOLE(LINE,1) = DIR                                            SHOLES     166
*       ..................................                            SHOLES     167
 300  PRINT*                                                          SHOLES     168
      PRINT*, 'ENTER "FROM" (E.G. ''02'' OR ''D1'')'                  SHOLES     169
      REWIND 1                                                        SHOLES     170
      READ(1,*,END=300) FROM                                          SHOLES     171
      LOK = 0                                                         SHOLES     172
      DOK = 0                                                         SHOLES     173
      NOK = 0                                                         SHOLES     174
      IF (GETLEN(FROM) .EQ. 2) THEN                                   SHOLES     175
        LOK = 1                                                       SHOLES     176
      END IF                                                          SHOLES     177
      IF (FROM(1:1) .EQ. 'D') THEN                                    SHOLES     178
        V = VAL(FROM(2:2))                                            SHOLES     179
        IF ((V .GE. 1) .AND. (V .LE. 6)) THEN                         SHOLES     180
          DOK = 1                                                     SHOLES     181
        END IF                                                        SHOLES     182
      END IF                                                          SHOLES     183
      IF ((ICHAR(FROM(1:1)) .GE. 16)                                  SHOLES     184
     +    .AND. (ICHAR(FROM(1:1)) .LE. 25)                            SHOLES     185
```

```
     +    .AND. (ICHAR(FROM(2:2)) .GE. 16)
     +    .AND. (ICHAR(FROM(2:2)) .LE. 25)
     +    .AND. (GETLEN(FROM) .EQ. 2)) THEN
          V = VAL(FROM)
          IF ((V .GE. 1) .AND. (V .LE. RMAX)) THEN
            NOK = 1
          END IF
        END IF
        IF ((LOK .EQ. 1) .AND. ((DOK .EQ. 1 ) .OR. (NOK .EQ. 1))) THEN
          HOLE(LINE,2) = FROM
        ELSE
          PRINT*
          PRINT*, 'INCORRECT ENTRY.  TRY AGAIN!!'
          GOTO 300
        END IF
*       ............................
  400     PRINT*
          PRINT*, 'ENTER "TO"  (E.G. ''02'' OR ''D1'')'
          REWIND 1
          READ(1,*,END=400) TO
          LOK = 0
          DOK = 0
          NOK = 0
          IF (GETLEN(TO) .EQ. 2) THEN
            LOK = 1
          END IF
          IF (TO(1:1) .EQ. 'D') THEN
            V = VAL(TO(2:2))
            IF ((V .GE. 1) .AND. (V .LE. 6)) THEN
              DOK = 1
            END IF
          END IF
          IF ((ICHAR(TO(1:1)) .GE. 16)
     +       .AND. (ICHAR(TO(1:1)) .LE. 25)
     +       .AND. (ICHAR(TO(2:2)) .GE. 16)
     +       .AND. (ICHAR(TO(2:2)) .LE. 25)
     +       .AND. (GETLEN(TO) .EQ. 2)) THEN
            V = VAL (TO)
            IF ((V .GE. 1) .AND. (V .LE. RMAX)) THEN
              NOK = 1
            END IF
          END IF
          IF ((LOK .EQ. 1) .AND. ((DOK .EQ. 1) .OR. (NOK .EQ. 1))) THEN
            HOLE (LINE,3) = TO
          ELSE
            PRINT*
            PRINT*, 'INCORRECT ENTRY.  TRY AGAIN!!'
            GOTO 400
          END IF
          IF(FROM .EQ. TO) THEN
            PRINT*
            PRINT*, 'INCORRECT ENTRY!!'
            PRINT*, '"FROM" CANNOT EQUAL "TO"'
            PRINT*, 'CHECK YOUR DATA AND REENTER "FROM" AND "TO"'
            PRINT*
            GOTO 300
          END IF
          IF ((FROM(1:1) .EQ. 'D' ) .AND. (TO(1:1) .EQ. 'D')) THEN
            PRINT*
            PRINT*,  'INCORRECT ENTRY!!'
            PRINT*, '"FROM" AND "TO" CANNOT BOTH CONTAIN "D"'
            PRINT*, '   CHECK YOUR DATA AND REENTER "FROM" AND "TO"'
            PRINT*
            GOTO 300
```

```
   129           END IF                                                           SHOLES      250
   130 *      .. ................................                                 SHOLES      251
   131 500    PRINT*                                                              SHOLES      252
   132        PRINT*, 'ENTER HOLE ''ID'' (E.G. ''WA'' OR ''DA'')'                 SHOLES      253
   133        REWIND 1                                                            SHOLES      254
   134        READ (1,*,END=500) ID                                               SHOLES      255
   135        IF ((( ID(1:1) .EQ. 'D' ) .OR. ( ID(1:1) .EQ. 'W' ))                SHOLES      256
   136       +   .AND. ( ICHAR(ID(2:2)) .LE. 58 )                                 SHOLES      257
   137       +   .AND. ( ICHAR(ID(2:2)) .GE. 33 )) THEN                           SHOLES      258
   138          HOLE(LINE,4) = ID                                                 SHOLES      259
   139        ELSE                                                                SHOLES      260
   140          GOTO 500                                                          SHOLES      261
   141        END IF                                                              SHOLES      262
   142 *      ................................                                    SHOLES      263
   143         RETURN                                                             SHOLES      264
   144         END                                                                SHOLES      265


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ANSWER   NONE              UNUSED/*S*       INTEGER
  DIR       727B                              CHAR*3
  DOK       724B                              INTEGER
  FROM      730B                              CHAR*3
  HERR        1B  /HOLEN/                     INTEGER
  HOLE        0B  /HOLEC/                     CHAR*3         140
  HTOT        0B  /HOLEN/                     INTEGER
  ID        732B                              CHAR*3
  INSERT      1   DUMMY-ARG UNUSED            INTEGER
  LINE        2   DUMMY-ARG                   INTEGER
  LOK       723B                              INTEGER
  NOK       725B                              INTEGER
  NROOMS   1244B  /ROOMN/                     INTEGER
  RAREA    1245B  /ROOMN/                     REAL            20
  ROOM        0B  /ROOMN/                     REAL           676
  TO        731B                              CHAR*3
  V         726B                              INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  HMAX    INTEGER                        35
  RMAX    INTEGER                        20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN    INTEGER        1      FUNCTION
  ICHAR     INTEGER        1      INTRINSIC
  VAL       INTEGER        1      FUNCTION


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

   200      7B                   34
   300     46B                   47
   400    203B                   81
   500    402B                  131
```

```
 SUBROUTINE DATAIN   74/175  OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DATAIN      5B     2


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH             735B =  477
  CM LABELLED COMMON LENGTH      1345B =  741
  CM STORAGE USED               61000B =  25088
  COMPILE TIME                  0.257 SECONDS
```

```
      SUBROUTINE MANIP (QUIT,ABORT)                                     SHOLES     266
************************************************************************COMH         1
***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH         2
************************************************************************COMH         3
      INTEGER HMAX                                                      COMH         4
      PARAMETER (HMAX = 35)                                             COMH         5
      COMMON /HOLEN/ HTOT, HERR                                         COMH         6
      COMMON /HOLEC/ HOLE(HMAX,4)                                       COMH         7
      INTEGER HTOT, HERR                                                COMH         8
      CHARACTER * 3 HOLE                                                COMH         9
* ==================================================                    COMH        10
* DESCRIPTION OF ARRAYS                                                 COMH        11
* ==================================================                    COMH        12
*     ROOM IDENTIFICATION            APERTURE ID                        COMH        13
* ----------------------------------                                    COMH        14
* DIRECTION    FROM ROOM     TO ROOM                                    COMH        15
* --------------------------------------------------                    COMH        16
* HOLE(X,1)    HOLE(X,2)     HOLE(X,3)    HOLE(X,4)                     COMH        17
*   A3           A3            A3           A3                          COMH        18
************************************************************************COMH        19
************************************************************************COMH        20
      INTEGER ABORT,ANSWER,DOK,FLAG1,LOK,N,NOK,OK,OK1,OK2,QUIT,INSERT   SHOLES     268
      INTEGER TEMP,V,X,Y,COMMAND                                        SHOLES     269
      CHARACTER * 3 DIR, FROM, TO                                       SHOLES     270
*                                                                       SHOLES     271
 10   FLAG1 = 0                                                         SHOLES     272
      PRINT*                                                            SHOLES     273
      PRINT*, 'CHOOSE'                                                  SHOLES     274
      PRINT*, '   (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES'      SHOLES     275
      PRINT*, '   (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA'   SHOLES     276
      PRINT*, '   (3) DELETE LINE           (6) STORE DATA AND EXIT ',  SHOLES     277
     +        'PROGRAM'                                                 SHOLES     278
      PRINT*, '                             (7) EXIT PROGRAM WITHOUT ', SHOLES     279
     +        'STORING DATA'                                            SHOLES     280
      PRINT*, 'ENTER A NUMBER !!!'                                      SHOLES     281
      PRINT*                                                            SHOLES     282
      REWIND 1                                                          SHOLES     283
      READ(1,*,END=10) COMMAND                                          SHOLES     284
*                                                                       SHOLES     285
*---------------------------------------------------------------------- SHOLES     286
*** DISPLAY LINE ***                                                    SHOLES     287
*---------------------------------------------------------------------- SHOLES     288
      IF ( COMMAND .EQ. 1 ) THEN                                        SHOLES     289
*                                                                       SHOLES     290
*** INDICATE EMPTY DATA FILE                                            SHOLES     291
        IF ( HTOT .EQ. 0 ) THEN                                         SHOLES     292
          PRINT*                                                        SHOLES     293
          PRINT*, 'DATA FILE IS EMPTY !!!'                              SHOLES     294
*                                                                       SHOLES     295
*** ENTER NUMBER OF LINE TO BE DISPLAYED                                SHOLES     296
        ELSE                                                            SHOLES     297
 100      PRINT*                                                        SHOLES     298
          PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED'      SHOLES     299
          PRINT*, '     ( ENTER "0" TO ESCAPE DISPLAY MODE )'           SHOLES     300
          REWIND 1                                                      SHOLES     301
          READ(1,*,END=100) N                                           SHOLES     302
*                                                                       SHOLES     303
*** CHECK VALIDITY OF LINE NUMBER                                       SHOLES     304
          IF ( (N .GT. HTOT) .OR. (N .LT. 0) ) THEN                     SHOLES     305
            PRINT*                                                      SHOLES     306
            PRINT*, 'INCORRECT NUMBER !!!!!!       TRY AGAIN !!!'       SHOLES     307
            PRINT*, '              -OR-  ENTER "0" TO ESCAPE FROM ',    SHOLES     308
     +              '"DISPLAY" MODE'                                    SHOLES     309
            GOTO 100                                                    SHOLES     310
```

```
*
*** ABORT 'DISPLAY' MODE
            ELSE IF ( N .EQ. 0 ) THEN
              PRINT*
              PRINT*, '     "DISPLAY" MODE ABORTED !!!'
*
*** DISPLAY LINE OF DATA
            ELSE IF ( (N .GT. 0) .AND. (N .LE. HTOT) ) THEN
              PRINT*
              CALL DISPLAY( N, COMMAND)
*
            END IF
          END IF
        END IF
*
*-------------------------------------------------------------------
*** INSERT LINE ***
*-------------------------------------------------------------------
        IF ( COMMAND .EQ. 2 ) THEN
*
*** INDICATE EMPTY DATA FILE
          IF ( HTOT .EQ. 0 ) THEN
            PRINT*
            PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** REQUEST NUMBER OF LINE BEFORE WHICH INSERTION IS TO BE MADE
          ELSE
200         PRINT*
            PRINT*, 'SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS ',
     +              'TO BE INSERTED'
            PRINT*, '      ( ENTER "0" TO ESCAPE "INSERTION" MODE )'
            REWIND 1
            READ(1,*,END=200) N
*
*** CHECK FOR VALID LINE NUMBER
            IF ( ( N .LT. 0 ) .OR. ( N .GT. HTOT ) ) THEN
              PRINT*
              PRINT*, 'INCORRECT LINE NUMBER !!!'
              PRINT*, '       TRY AGAIN !!!  -OR-  ENTER "0" TO ESCAPE',
     +                '"INSERTION" MODE'
              GOTO 200
*
*** ABORT INSERTION MODE
            ELSE IF ( N .EQ. 0 ) THEN
              PRINT*
              PRINT*, '       "INSERTION" MODE ABORTED'
*
*** MAKE ROOM FOR NEW LINE OF DATA
            ELSE IF ( (N .GT. 0) .AND. (N .LE. HTOT) ) THEN
              DO 230 X = HTOT,N,-1
                DO 210 Y = 1,4
                  HOLE(X+1,Y) = HOLE(X,Y)
210             CONTINUE
230           CONTINUE
*
*** ENTER DATA FOR NEW LINE
              HTOT = HTOT + 1
              CALL DATAIN (1,N)
*
                PRINT*
                PRINT*, 'THE FOLLOWING LINE HAS BEEN ADDED AS LINE ', N
                CALL DISPLAY( N, COMMAND)
            END IF
          END IF
```

```
      END IF
*
*---------------------------------------------------------------------
*** DELETE LINE ***
*---------------------------------------------------------------------
      IF ( COMMAND .EQ. 3 ) THEN
*
*** INDICATE EMPTY DATA FILE
        IF ( HTOT .EQ. 0 ) THEN
          PRINT*
          PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** READ NUMBER OF LINE TO BE DELETED
        ELSE
300       PRINT*
          PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DELETED'
          PRINT*, '   (ENTER "0" TO ESCAPE DELETION MODE)'
          REWIND 1
          READ(1,*,END=300) N
*
*** CHECK VALIDITY OF LINE NUMBER
          IF ( (N .GT. HTOT ) .OR. ( N .LT. 0 ) ) THEN
            PRINT*
            PRINT*, ' INCORRECT NUMBER !!!'
            PRINT*, '    TRY AGAIN !!!  -OR-  ENTER "0" TO ESCAPE FROM',
     +              '"DELETE" MODE'
            GOTO 300
*
*** ABORT 'DELETE' MODE
          ELSE IF ( N .EQ. 0 ) THEN
            PRINT*, '  "DELETE" MODE ABORTED'
*
*** DOUBLE CHECK CHOICE OF LINE TO BE DELETED
          ELSE IF (( N .GT. 0 ) .AND. ( N .LE. HTOT )) THEN
            PRINT*
            PRINT*, 'DOUBLE CHECK !!!'
            PRINT*, '     DO YOU WANT TO DELETE THE FOLLOWING LINE?:'
            CALL DISPLAY( N, COMMAND)
305         PRINT*, '   ENTER (1) YES  OR   (2) NO'
            REWIND 1
            READ(1,*,END=305) ANSWER
*
*** DELETE LINE
            IF ( ANSWER .EQ. 1 ) THEN
              DO 330 X = N, HTOT - 1
                DO 310 Y = 1,4
                  HOLE(X,Y) = HOLE(X+1,Y)
310             CONTINUE
330           CONTINUE
              HTOT = HTOT - 1
              PRINT*
              PRINT*, 'LINE # ',N,' DELETED'
            END IF
*
          END IF
        END IF
      END IF
*
*---------------------------------------------------------------------
*** DISPLAY ALL DATA ***
*---------------------------------------------------------------------
      IF ( COMMAND .EQ. 4 ) THEN
*
*** INDICATE EMPTY DATA FILE
```

```
          IF ( HTOT .EQ. 0 ) THEN
            PRINT*
            PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** DISPLAY DATA
          ELSE
            PRINT*
            CALL DISPLAY( N, COMMAND)
*
          END IF
        END IF
*
*-----------------------------------------------------------------------
*** ADD DATA ***
*-----------------------------------------------------------------------
        IF ( COMMAND .EQ. 5 ) THEN
*
*** ENTER DATA
500       HTOT = HTOT + 1
          CALL DATAIN (0,HTOT)
510       PRINT*
          PRINT*, 'DO YOU WANT TO ENTER MORE DATA?   (1) YES   (2) NO'
          PRINT*, '   ENTER A NUMBER !!!'
          REWIND 1
          READ(1,*,END=510) ANSWER
*
*** CHECK VALIDITY OF NUMBER
          IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN
            GOTO 510
*
*** ENTER MORE DATA
          ELSE IF ( ANSWER .EQ. 1 ) THEN
            GOTO 500
*
*** DISCONTINUE DATA ENTRY
          ELSE IF ( ANSWER .EQ. 2 ) THEN
            PRINT*
            PRINT*, 'DATA ENTRY DISCONTINUED'
*
          END IF
        END IF
*
*-----------------------------------------------------------------------
*** STORE DATA AND PROGRAM ***
*-----------------------------------------------------------------------
        IF ( COMMAND .EQ. 6 ) THEN
 600      PRINT*
          PRINT*, 'DOUBLE CHECK !!!'
          PRINT*, '     DO YOU YOU WANT TO STORE THIS DATA AND END PROG'
          PRINT*, '  NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE '
          PRINT*, '  OF THE SAME NAME !!!'
          PRINT*, '         ENTER A NUMBER: (1) YES   (2) NO'
          REWIND 1
          READ(1,*,END=600) ANSWER
*
*** SET FLAG FOR STORING DATA IN THE MAIN PROGRAM
          IF ( ANSWER .EQ. 1 ) THEN
            QUIT = 1
            RETURN
*
*** ABORT 'STORING' MODE
          ELSE IF ( ANSWER .EQ. 2 ) THEN
            PRINT*
            PRINT*, '    "STORING" MODE DISCONTINUED'
```

```
*                                                                          SHOLES     503
*** CHECK VALIDITY OF ANSWER                                               SHOLES     504
          ELSE IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN       SHOLES     505
            GOTO 600                                                       SHOLES     506
*                                                                          SHOLES     507
          END IF                                                           SHOLES     508
        END IF                                                             SHOLES     509
*                                                                          SHOLES     510
*------------------------------------------------------------------------  SHOLES     511
*** END PROGRAM WITHOUT STORING DATA ***                                   SHOLES     512
*------------------------------------------------------------------------  SHOLES     513
        IF ( COMMAND .EQ. 7 ) THEN                                         SHOLES     514
 700      PRINT*                                                           SHOLES     515
          PRINT*, 'DOUBLE CHECK !!!'                                       SHOLES     516
          PRINT*, '    DO YOU WANT TO END THIS PROGRAM ',                  SHOLES     517
     +         'WITHOUT STORING DATA?'                                     SHOLES     518
          PRINT*, '        ENTER A NUMBER: (1) YES   (2) NO'               SHOLES     519
          REWIND 1                                                         SHOLES     520
          READ(1,*,END=700) ANSWER                                         SHOLES     521
*                                                                          SHOLES     522
*** SET FLAG FOR ABORTING PROGRAM IN THE MAIN PROGRAM                      SHOLES     523
          IF ( ANSWER .EQ. 1 ) THEN                                        SHOLES     524
            ABORT = 1                                                      SHOLES     525
            RETURN                                                         SHOLES     526
*                                                                          SHOLES     527
*** ABORT 'STORING' MODE                                                   SHOLES     528
          ELSE IF ( ANSWER .EQ. 2 ) THEN                                   SHOLES     529
            PRINT*                                                         SHOLES     530
            PRINT*, '    "ABORTION" MODE DISCONTINUED'                     SHOLES     531
*                                                                          SHOLES     532
*** CHECK VALIDITY OF ANSWER                                               SHOLES     533
          ELSE IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN       SHOLES     534
            GOTO 700                                                       SHOLES     535
*                                                                          SHOLES     536
          END IF                                                           SHOLES     537
        END IF                                                             SHOLES     538
*                                                                          SHOLES     539
*------------------------------------------------------------------------  SHOLES     540
*** LOOP TO BEGINNING OF 'MANIP' SUBROUTINE                                SHOLES     541
*------------------------------------------------------------------------  SHOLES     542
        GOTO 10                                                            SHOLES     543
*                                                                          SHOLES     544
        RETURN                                                             SHOLES     545
TRIVIAL* NO PATH TO THIS STATEMENT
        END                                                                SHOLES     546
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ABORT       2   DUMMY-ARG                  INTEGER
  ANSWER   1400B                             INTEGER
  COMMAND  1405B                             INTEGER
  DIR      NONE             UNUSED/*S*       CHAR*3
  DOK      NONE             UNUSED/*S*       INTEGER
  FLAG1    1401B                             INTEGER
  FROM     NONE             UNUSED/*S*       CHAR*3
  HERR       1B   /HOLEN/                    INTEGER
  HOLE       0B   /HOLEC/                    CHAR*3       140
  HTOT       0B   /HOLEN/                    INTEGER
  INSERT   NONE             UNUSED/*S*       INTEGER
  LOK      NONE             UNUSED/*S*       INTEGER
  N        1402B                             INTEGER
  NOK      NONE             UNUSED/*S*       INTEGER
```

```
 OK        NONE                 UNUSED/*S*        INTEGER
 OK1       NONE                 UNUSED/*S*        INTEGER
 OK2       NONE                 UNUSED/*S*        INTEGER
 QUIT         1    DUMMY-ARG                      INTEGER
 TEMP      NONE                 UNUSED/*S*        INTEGER
 TO        NONE                 UNUSED/*S*        CHAR*3
 V         NONE                 UNUSED/*S*        INTEGER
 X         1403B                                  INTEGER
 Y         1404B                                  INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  HMAX    INTEGER                         35


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  DATAIN                   2      SUBROUTINE
  DISPLAY                  2      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    10       7B                    26          310  INACTIVE   DO-TERM        176
   100      50B                    52          330  INACTIVE   DO-TERM        177
   200     133B                    92          500    437B                    211
   210  INACTIVE   DO-TERM        117          510    443B                    213
   230  INACTIVE   DO-TERM        118          600    504B                    239
   300     263B                   143          700    556B                    269
   305     332B                   167


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  MANIP       5B     2


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH           1414B =  780
  CM LABELLED COMMON LENGTH       54B =  44
  CM STORAGE USED             63000B =  26112
  COMPILE TIME                0.388 SECONDS

      1  TRIVIAL   ERROR IN  MANIP
```

```
  1       SUBROUTINE DISPLAY (LINE, COMMAND)                              SHOLES   547
  2 ************************************************************************COMH      1
  3 ***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH      2
  4 ************************************************************************COMH      3
  5       INTEGER HMAX                                                    COMH      4
  6       PARAMETER (HMAX = 35)                                           COMH      5
  7       COMMON /HOLEN/ HTOT, HERR                                       COMH      6
  8       COMMON /HOLEC/ HOLE(HMAX,4)                                     COMH      7
  9       INTEGER HTOT, HERR                                              COMH      8
 10       CHARACTER * 3 HOLE                                              COMH      9
 11 * ==================================================               COMH     10
 12 * DESCRIPTION OF ARRAYS                                             COMH     11
 13 * ==================================================               COMH     12
 14 *     ROOM IDENTIFICATION            APERTURE ID                    COMH     13
 15 * ------------------------------------                              COMH     14
 16 * DIRECTION    FROM ROOM     TO ROOM                                COMH     15
 17 * --------------------------------------------------               COMH     16
 18 * HOLE(X,1)    HOLE(X,2)     HOLE(X,3)    HOLE(X,4)                 COMH     17
 19 *  A3             A3            A3           A3                     COMH     18
 20 ************************************************************************COMH     19
 21 ************************************************************************COMH     20
 22       INTEGER LINE, COMMAND, N                                        SHOLES   549
 23 1000  FORMAT (8(3X,A))                                                SHOLES   550
 24 2000  FORMAT (4X,I3,8X,A3,7X,A3,3X,A3,2X,A3)                          SHOLES   551
 25       PRINT 1000, 'LINE #','DIRECTION','FROM','TO','ID'               SHOLES   552
 26       IF ( COMMAND .EQ. 4 ) THEN                                      SHOLES   553
 27         DO 10 N = 1,HTOT                                              SHOLES   554
 28           PRINT 2000, N,HOLE(N,1),HOLE(N,2),HOLE(N,3),HOLE(N,4)       SHOLES   555
 29 10      CONTINUE                                                      SHOLES   556
 30       ELSE                                                            SHOLES   557
 31           PRINT 2000, LINE,HOLE(LINE,1),HOLE(LINE,2),HOLE(LINE,3),    SHOLES   558
 32      +          HOLE(LINE,4)                                          SHOLES   559
 33       END IF                                                          SHOLES   560
 34       RETURN                                                          SHOLES   561
 35       END                                                             SHOLES   562
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 COMMAND     2   DUMMY-ARG                   INTEGER
 HERR       1B   /HOLEN/                     INTEGER
 HOLE       0B   /HOLEC/                     CHAR*3        140
 HTOT       0B   /HOLEN/                     INTEGER
 LINE        1   DUMMY-ARG                   INTEGER
 N        204B                               INTEGER
```

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 HMAX    INTEGER                          35
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM       29
  1000    123B     FORMAT        23
  2000    125B     FORMAT        24
```



```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 DISPLAY     5B     2
```

```
--STATISTICS--

 PROGRAM-UNIT LENGTH            210B =  136
 CM LABELLED COMMON LENGTH       54B =  44
 CM STORAGE USED              61000B =  25088
 COMPILE TIME                 0.068 SECONDS
```

```
      INTEGER FUNCTION VAL(STRING)                               SHOLES     563
C** RETURNS THE INTEGER VALUE OF A STRING.                       SHOLES     564
      INTEGER NUMBER, X,L,EXP,DIGIT,GETLEN                       SHOLES     565
      CHARACTER * (*) STRING                                     SHOLES     566
      L = GETLEN(STRING)                                         SHOLES     567
      NUMBER = 0                                                 SHOLES     568
      DO 10 X = L,1,-1                                           SHOLES     569
      EXP = L - X                                                SHOLES     570
      DIGIT = ICHAR(STRING(X:X)) - 16                            SHOLES     571
      NUMBER = NUMBER + DIGIT*10**EXP                            SHOLES     572
10    CONTINUE                                                   SHOLES     573
      VAL = NUMBER                                               SHOLES     574
      RETURN                                                     SHOLES     575
      END                                                        SHOLES     576
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 DIGIT     76B                              INTEGER
 EXP       75B                              INTEGER
 L         74B                              INTEGER
 NUMBER    72B                              INTEGER
 STRING     1   DUMMY-ARG                   CHAR*(*)
 VAL       71B                              INTEGER
 X         73B                              INTEGER


--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 GETLEN    INTEGER        1      FUNCTION
 ICHAR     INTEGER        1      INTRINSIC


--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        11


--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 VAL         6B     1


--STATISTICS--

 PROGRAM-UNIT LENGTH           102B = 66
 CM STORAGE USED             61000B = 25088
 COMPILE TIME                0.039 SECONDS
```

```
      INTEGER FUNCTION GETLEN (STRING)                          SHOLES   577
C                                                               SHOLES   578
C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING        SHOLES   579
C                                                               SHOLES   580
C                                                               SHOLES   581
C ARGUMENT DEFINITIONS --                                       SHOLES   582
C   READ ARGUMENTS                                              SHOLES   583
C      STRING - STRING WHOSE LENGTH IS TO BE DETERMINED         SHOLES   584
C                                                               SHOLES   585
      CHARACTER * (*) STRING                                    SHOLES   586
C                                                               SHOLES   587
C FUNCTION PARAMETERS                                           SHOLES   588
      CHARACTER * 1 BLANK                                       SHOLES   589
      PARAMETER (BLANK = ' ')                                   SHOLES   590
C                                                               SHOLES   591
C LOCAL VARIABLES                                               SHOLES   592
      INTEGER NEXT                                              SHOLES   593
C                                                               SHOLES   594
C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK    SHOLES   595
      DO 10 NEXT = LEN(STRING),1,-1                             SHOLES   596
         IF (STRING(NEXT : NEXT) .NE. BLANK) THEN               SHOLES   597
            GETLEN = NEXT                                       SHOLES   598
            RETURN                                              SHOLES   599
         END IF                                                 SHOLES   600
   10 CONTINUE                                                  SHOLES   601
C                                                               SHOLES   602
C ALL CHARACTERS ARE BLANKS                                     SHOLES   603
      GETLEN = 0                                                SHOLES   604
C                                                               SHOLES   605
      RETURN                                                    SHOLES   606
      END                                                       SHOLES   607
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 GETLEN     63B                               INTEGER
 NEXT       64B                               INTEGER
 STRING      1   DUMMY-ARG                    CHAR*(*)


--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE------------------------VALUE

 BLANK   CHAR*1                          ' '


--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 LEN       INTEGER         1      INTRINSIC


--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        25
```

```
FTN 5.1+552          83/12/24. 09.11.45 PAGE     20
  FUNCTION GETLEN    74/175  OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 GETLEN      6B      1


--STATISTICS--

 PROGRAM-UNIT LENGTH              70B =  56
 CM STORAGE USED              61000B =  25088
 COMPILE TIME                 0.037 SECONDS
```

```
      SUBROUTINE LHOLE                                                  LHOLE     1
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LHOLE     2
*!!!                                                                 !!!LHOLE     3
*!!!   LOAD THE CONTENTS OF THE "HOLE" FILE INTO THE "HOLE" ARRAY    !!!LHOLE     4
*!!!                                                                 !!!LHOLE     5
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LHOLE     6
************************************************************************COMH      1
***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH      2
************************************************************************COMH      3
      INTEGER HMAX                                                      COMH      4
      PARAMETER (HMAX = 35)                                             COMH      5
      COMMON /HOLEN/ HTOT, HERR                                         COMH      6
      COMMON /HOLEC/ HOLE(HMAX,4)                                       COMH      7
      INTEGER HTOT, HERR                                                COMH      8
      CHARACTER * 3 HOLE                                                COMH      9
* ==================================================                    COMH     10
* DESCRIPTION OF ARRAYS                                                 COMH     11
* ==================================================                    COMH     12
*     ROOM IDENTIFICATION              APERTURE ID                      COMH     13
* -----------------------------------                                   COMH     14
* DIRECTION    FROM ROOM     TO ROOM                                    COMH     15
* --------------------------------------------------                    COMH     16
* HOLE(X,1)    HOLE(X,2)     HOLE(X,3)    HOLE(X,4)                     COMH     17
*   A3            A3            A3           A3                         COMH     18
************************************************************************COMH     19
************************************************************************COMH     20
************************************************************************COMF      1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF      2
************************************************************************COMF      3
      INTEGER FMAX                                                      COMF      4
      PARAMETER (FMAX = 50)                                             COMF      5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF      6
     $                 FTOT                                             COMF      7
      COMMON /INITILC/ BLDG                                             COMF      8
      CHARACTER * 5 BLDG                                                COMF      9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF     10
      INTEGER QUALITY, FERR, FTOT                                       COMF     11
************************************************************************COMF     12
************************************************************************COMF     13
      INTEGER GETLEN, R, C                                              LHOLE     9
      CHARACTER * 7  PFN                                                LHOLE    10
      PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'H'                          LHOLE    11
      HERR = 0                                                          LHOLE    12
      CALL PF  ('GET',0,PFN(1:GETLEN(PFN)),'RC',HERR)                   LHOLE    13
      IF ( HERR .EQ. 0 ) THEN                                           LHOLE    14
        OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',                       LHOLE    15
     $        STATUS='OLD', ACCESS='SEQUENTIAL')                        LHOLE    16
1000     FORMAT (1X,4(1X,A3))                                           LHOLE    17
        HTOT = 0                                                        LHOLE    18
        DO 10 R = 1,HMAX                                                LHOLE    19
          READ (3,1000,END=20)(HOLE(R,C),C=1,4)                         LHOLE    20
          HTOT = HTOT + 1                                               LHOLE    21
10      CONTINUE                                                        LHOLE    22
20      CONTINUE                                                        LHOLE    23
        CLOSE(3,STATUS='DELETE')                                        LHOLE    24
      ELSE IF ( HERR .EQ. 2 ) THEN                                      LHOLE    25
        CALL WARNING (1)                                                LHOLE    26
      ELSE                                                              LHOLE    27
        CALL WARNING (2)                                                LHOLE    28
      END IF                                                            LHOLE    29
      RETURN                                                            LHOLE    30
      END                                                               LHOLE    31
```

```
SUBROUTINE LHOLE     74/175  OPT=0

--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 AFLAG        2B  /INITILN/                 REAL
 BLDG         0B  /INITILC/                 CHAR*5
 C          214B                            INTEGER
 FERR        66B  /INITILN/                 INTEGER
 FREQ         0B  /INITILN/                 REAL
 FREQA        4B  /INITILN/                 REAL            50
 FTOT        67B  /INITILN/                 INTEGER
 HERR         1B  /HOLEN/                   INTEGER
 HOLE         0B  /HOLEC/                   CHAR*3         140
 HTOT         0B  /HOLEN/                   INTEGER
 PFN        215B                            CHAR*7
 QUALITY      1B  /INITILN/                 INTEGER
 R          213B                            INTEGER
 RFLAG        3B  /INITILN/                 REAL


--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE------------------------VALUE

 FMAX    INTEGER                        50
 HMAX    INTEGER                        35


--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 GETLEN    INTEGER         1      FUNCTION
 PF                        5      SUBROUTINE
 WARNING                   1      SUBROUTINE


--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        53
    20     73B                    54
  1000    130B     FORMAT         48


--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 LHOLE       5B     0


--I/O UNITS--(LO=A)
-NAME--- PROPERTIES------------

 TAPE3   AUX/FMT/SEQ


--STATISTICS--

 PROGRAM-UNIT LENGTH           222B =  146
 CM LABELLED COMMON LENGTH     145B =  101
 CM STORAGE USED             61000B =  25088
 COMPILE TIME                0.086 SECONDS
```

```
      SUBROUTINE ERROR(IERR)                                            ERROR      1
      CHARACTER*45 MESSAGE(20)                                          ERROR      2
      DATA MESSAGE( 1)/'MATERIALS DATA BASE IS EMPTY                 '/ ERROR      3
      DATA MESSAGE( 2)/'FREQUENCY IS OUT OF RANGE                    '/ ERROR      4
      DATA MESSAGE( 3)/'THIS MATERIAL IS NOT IN DATA BASE            '/ ERROR      5
      DATA MESSAGE( 4)/'DENOMINATOR IS ZERO                          '/ ERROR      6
      DATA MESSAGE( 5)/'FILE HANDLING ERROR                          '/ ERROR      7
      DATA MESSAGE( 6)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR      8
      DATA MESSAGE( 7)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR      9
      DATA MESSAGE( 8)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     10
      DATA MESSAGE( 9)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     11
      DATA MESSAGE(10)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     12
      DATA MESSAGE(11)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     13
      DATA MESSAGE(12)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     14
      DATA MESSAGE(13)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     15
      DATA MESSAGE(14)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     16
      DATA MESSAGE(15)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     17
      DATA MESSAGE(16)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     18
      DATA MESSAGE(17)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     19
      DATA MESSAGE(18)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     20
      DATA MESSAGE(19)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     21
      DATA MESSAGE(20)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR     22
      IERRM=5                                                           ERROR     23
      IF(IERR.GT.IERRM) IERR=20                                         ERROR     24
      WRITE(6,10) IERR,MESSAGE(IERR)                                    ERROR     25
   10 FORMAT(' ***ERROR NUMBER = ',I5,' ***  ',A45)                     ERROR     26
      CALL PMDSTOP                                                      ERROR     27
      STOP 'ERROR'                                                      ERROR     28
      END                                                               ERROR     29
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  IERR   .     1   DUMMY-ARG                  INTEGER
  IERRM      210B                             INTEGER
  MESSAGE     56B                             CHAR*45          20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  PMDSTOP                    0      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10      36B     FORMAT         26


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ERROR       5B     1
```

```
FTN 5.1+552          83/12/24. 09.11.45 PAGE     24
SUBROUTINE ERROR     74/175  OPT=0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------

  TAPE6   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH              213B =   139
  CM STORAGE USED                61000B = 25088
  COMPILE TIME                   0.055 SECONDS
```

```
       SUBROUTINE WARNING(ERR)                                          WARNING      1
       INTEGER ERR, ERRM                                                WARNING      2
       CHARACTER*45 MESSAGE(20)                                         WARNING      3
       DATA MESSAGE( 1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING     4
       DATA MESSAGE( 2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE    '/ WARNING     5
       DATA MESSAGE( 3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG   '/ WARNING     6
       DATA MESSAGE( 4)/'FILE HANDLING PROBLEM ON "MATTER FILE        '/ WARNING     7
       DATA MESSAGE( 5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING     8
       DATA MESSAGE( 6)/'FILE HANDLING PROBLEM ON "TYPE" FILE         '/ WARNING     9
       DATA MESSAGE( 7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING    10
       DATA MESSAGE( 8)/'FILE HANDLING PROBLEM ON "WALL" FILE         '/ WARNING    11
       DATA MESSAGE( 9)/'HEIGHT AND WIDTH OF ROOM MISSING             '/ WARNING    12
       DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING                    '/ WARNING    13
       DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG       '/ WARNING    14
       DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE         '/ WARNING    15
       DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING    16
       DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING    17
       DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING    18
       DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING    19
       DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING    20
       DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING    21
       DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING    22
       DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING    23
       ERRM=12                                                          WARNING     24
       IERR = ERR                                                       WARNING     25
       IF(ERR.GT.ERRM) IERR=20                                          WARNING     26
       WRITE(6,20)                                                      WARNING     27
       WRITE(6,10) ERR,MESSAGE(IERR)                                    WARNING     28
       WRITE(6,20)                                                      WARNING     29
 10    FORMAT(' ***WARNING NUMBER = ',I5,' ***  ',A45)                  WARNING     30
 20    FORMAT('      ')                                                 WARNING     31
       RETURN                                                           WARNING     32
       END                                                              WARNING     33
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 ERR         1   DUMMY-ARG                   INTEGER
 ERRM      60B                               INTEGER
 IERR     213B                               INTEGER
 MESSAGE   61B                               CHAR*45        20


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10     34B     FORMAT         30
    20     42B     FORMAT         31


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 WARNING     5B     1
```

```
FTN 5.1+552        83/12/24. 09.11.45 PAGE     26
SUBROUTINE WARNING  74/175  OPT=0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

 TAPE6   FMT/SEQ


--STATISTICS--

 PROGRAM-UNIT LENGTH              216B =   142
 CM STORAGE USED                61000B =  25088
 COMPILE TIME                   0.057 SECONDS
```

Appendix 9.4 Listing of Computer Program SWALLS

```
      PROGRAM SWALLS (INPUT,TAPE1=INPUT)                                SWALLS
*                                                                       SWALLS
*THIS INTERACTIVE PROGRAM INPUTS THE DATA DESCRIBING EACH WALL          SWALLS
*IN THE BUILDING AND STORES IT.  THE FILE NAME IS CREATED BY            SWALLS
*ATTACHING "B" TO THE FRONT OF AND "W" TO THE BACK OF THE BUILDING      SWALLS
*IDENTIFICATION. THE BUILDING IDENTIFICATION CAN BE NO MORE             SWALLS
*THAN 5 ALPHANUMERIC CHARACTERS.                                        SWALLS
                                                                        SWALLS
************************************************************************COMF
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF
************************************************************************COMF
      INTEGER FMAX                                                      COMF
      PARAMETER (FMAX = 50)                                             COMF
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF
     $                 FTOT                                             COMF
      COMMON /INITILC/ BLDG                                             COMF
      CHARACTER * 5 BLDG                                                COMF
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF
      INTEGER QUALITY, FERR, FTOT                                       COMF
************************************************************************COMF
************************************************************************COMF
************************************************************************COMR
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR
************************************************************************COMR
      INTEGER RMAX                                                      COMR
      PARAMETER (RMAX = 20)                                             COMR
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR
      INTEGER NROOMS                                                    COMR
      REAL ROOM                                                         COMR
************************************************************************COMR
************************************************************************COMR
************************************************************************COMW
***   COMMON FOR DATABASE OF WALL PARAMETERS                         ***COMW
************************************************************************COMW
      INTEGER WMAX                                                      COMW
      PARAMETER (WMAX = 75)                                             COMW
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW
      COMMON /WALLC/ WALL(WMAX,4)                                       COMW
      INTEGER WTOT,WERR                                                 COMW
      REAL WDIM                                                         COMW
      CHARACTER *3 WALL                                                 COMW
* ================================                                      COMW
** DESCRIPTION OF ARRAYS                                                COMW
* ================================                                      COMW
*    WALL IDENTIFICATION                                                COMW
* --------------------------------                                      COMW
* DIRECTION    FROM       TO                                            COMW
*              ROOM       ROOM                                          COMW
* --------------------------------                                      COMW
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                     COMW
*  A3          A3          A3                                           COMW
* ======================================================                COMW
*        WALL PARAMETERS                                                COMW
* ------------------------------------------------------                COMW
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                   COMW
* ------------------------------------------------------                COMW
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                         COMW
*  A3          F8.2        F8.2        F8.2                             COMW
************************************************************************COMW
************************************************************************COMW
      INTEGER GETLEN,QUIT,ABORT,ANSWER,OLDFILE,N,Y1,Y2,LINE             SWALLS
      INTEGER IERR                                                      SWALLS
      CHARACTER * 7  PFN                                                SWALLS
*                                                                       SWALLS
```

```
* INITIALIZATION
      QUIT = 0
      WTOT = 0
      ABORT = 0
100   PRINT*
      PRINT *, 'ENTER BUILDING IDENTIFICATION  (E.G. ''101'')'
      PRINT *, '    (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)'
      REWIND 1
      READ(1,*,END=100) BLDG

      IF ( GETLEN(BLDG) .GT. 5 ) THEN
        GO TO 100
      END IF
      PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'W'
*
*** LOAD DATA FROM EXISTING FILE IF NECESSARY
200   PRINT*
      PRINT*,'WILL THIS BE'
      PRINT*,'    (1) A MODIFICATION OF AN EXISTING FILE?'
      PRINT*,'    (2) A NEW FILE?'
      PRINT*,'ENTER A NUMBER  !!!'
      REWIND 1
      READ(1,*,END=200) OLDFILE
      IF ( ( OLDFILE .NE. 1 ) .AND. ( OLDFILE .NE. 2 ) ) THEN
        GOTO 200
      ELSE IF ( OLDFILE .EQ. 1 ) THEN
*
*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME
        IERR = 0
        CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)
        IF ( IERR .EQ. 2 ) THEN
          PRINT*
          PRINT *, 'FILE ',PFN, ' NOT FOUND'
          PRINT*, 'PROGRAM ABORTED!!!'
          PRINT*
          PRINT*, 'FIND CORRECT BUILDING IDENTIFIER AND RESTART ',
     +              'PROGRAM'
          PRINT*
          STOP
*
        ELSE
          CALL LWALL
          IF (WERR .NE. 0) CALL ERROR(5)
        ENDIF
      ELSE IF ( OLDFILE .EQ. 2 ) THEN
*
*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME
        IERR = 0
        CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)
        IF ( IERR .EQ. 0 ) THEN
          PRINT*
          PRINT*, 'DATA FILE ALREADY EXISTS FOR BUILDING ',BLDG
          PRINT*
          PRINT*,'IF YOU ENTER DATA AND STORE IT, YOU WILL WRITE ',
     +             'OVER THE OLD FILE.'
250       PRINT*
          PRINT*,'YOU MAY EITHER (1) ABORT OR (2) CONTINUE.'
          PRINT*,'INDICATE YOUR CHOICE BY ENTERING A NUMBER.'
          REWIND 1
          READ(1,*,END=250) ANSWER
            PRINT*
            PRINT*,'PROGRAM HAS BEEN ABORTED, PER YOUR REQUEST'
            PRINT*
          IF ( ANSWER .EQ. 1 ) THEN
```

```
            STOP
          ELSE IF ( ANSWER .EQ. 2 ) THEN
 9090       CONTINUE
          ELSE
            GOTO 250
          ENDIF
        ELSE IF ( IERR .EQ. 2 ) THEN
*
*         NO DATA FILE EXISTS FOR THIS BUILDING AND DATA ENTRY
*         CAN CONTINUE
*
 9091     CONTINUE
        ELSE
*
*         **PERMANENT FILE ERROR
*
          PRINT*
          PRINT*,'PROGRAM ABORTED !!!'
          PRINT*,'  SOME PERMANENT FILE ERROR HAS OCCURRED'
          PRINT*,'  DOUBLE CHECK YOUR BUILDING IDENTIFICATION ',
     +             'AND TRY AGAIN'
          STOP
        ENDIF
        PRINT*
        PRINT*, ' BEGIN ENTERING DATA'
****************************************************************
*   CHECK TO SEE IF THERE IS ENOUGH ARRAY SPACE
****************************************************************
 300   IF(WTOT.GE.WMAX) THEN
       WTOT = WMAX
       PRINT *, 'DATA ENTRY ABORTED.'
       PRINT *, 'MAXIMUM NUMBER OF DATA LINES IN FILE WOULD'
       PRINT *, ' HAVE BEEN EXCEEDED. NO MORE THAN ',WMAX
       PRINT *, ' DATA LINES ARE ALLOWED.'
       PRINT *, ' TO INCREASE THE MAXIMUM NUMBER OF ENTRIES ALLOWED,'
       PRINT *, ' CHANGE THE PARAMETER "WMAX" IN EACH COMMON OF'
       PRINT *, ' EVERY SUBROUTINE (THERE ARE FOUR PLACES).'
       PRINT *, ' THEN RECOMPILE THE PROGRAM.'
       GOTO 450
       ENDIF
****************************************************************
        WTOT = WTOT + 1
        IF ( WTOT .EQ. 1) THEN
          CALL DATAIN(1,WTOT)
        ELSE
          CALL DATAIN (0,WTOT)
        END IF
 400    PRINT*
        PRINT*, 'DO YOU WANT TO ENTER MORE DATA?',
     +             '(1) YES  (2) NO'
        PRINT*, '  ENTER A NUMBER !!!'
        REWIND 1
        READ(1,*,END=400) ANSWER
        IF ( (ANSWER .NE. 1) .AND. (ANSWER .NE. 2) ) THEN
          GOTO 400
        ELSE IF ( ANSWER .EQ. 1) THEN
          GOTO 300
        ELSE IF ( ANSWER .EQ. 2 ) THEN
          PRINT*
          PRINT*, 'DATA ENTRY DISCONTINUED'
        END IF
       END IF
*
*** MANIPULATE DATA
```

```
 450   CALL MANIP (QUIT,ABORT)
 *
 *** TERMINATE PROGRAM, STORING DATA IF NECESSARY
       IF ( QUIT .EQ. 1 ) THEN
        OPEN(UNIT=6,FILE=PFN(1:GETLEN(PFN)),FORM='FORMATTED',
     +           ACCESS='SEQUENTIAL',STATUS='NEW')
 500     FORMAT (1X,4(1X,A3),3(1X,F8.2))
         DO 600 N = 1,WTOT
           WRITE (6,500)(WALL(N,Y1), Y1=1,4),(WDIM(N,Y2), Y2=1,3)
 600     CONTINUE
         ENDFILE(6)
         CALL PF ('REPLACE',0,PFN(1:GETLEN(PFN)))
WARNING* NUMBER OF ARGUMENTS IN REFERENCE TO _PF IS NOT CONSISTENT
         CLOSE(6,STATUS='DELETE')
         PRINT*
         PRINT*,'DATA HAS BEEN STORED AND PROGRAM TERMINATED'
       END IF
       IF( ABORT .EQ. 1 ) THEN
         PRINT*
         PRINT*, 'PROGRAM HAS BEEN ABORTED'
         PRINT*,'      NO DATA HAS BEEN STORED !!!'
       END IF
       STOP
       END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ABORT    1216B                             INTEGER
  AFLAG       2B  /INITILN/                  REAL
  ANSWER   1217B                             INTEGER
  BLDG        0B  /INITILC/                  CHAR*5
  FERR       66B  /INITILN/                  INTEGER
  FREQ        0B  /INITILN/                  REAL
  FREQA       4B  /INITILN/                  REAL            50
  FTOT       67B  /INITILN/                  INTEGER
  IERR     1224B                             INTEGER
  LINE     NONE             UNUSED/*S*       INTEGER
  N        1221B                             INTEGER
  NROOMS   1244B  /ROOMN/                    INTEGER
  OLDFILE  1220B                             INTEGER
  PFN      1225B                             CHAR*7
  QUALITY     1B  /INITILN/                  INTEGER
  QUIT     1215B                             INTEGER
  RAREA    1245B  /ROOMN/                    REAL            20
  RFLAG       3B  /INITILN/                  REAL
  ROOM        0B  /ROOMN/                    REAL           676
  WALL        0B  /WALLC/                    CHAR*3         300
  WDIM        0B  /WALLN/                    REAL           225
  WERR      342B  /WALLN/                    INTEGER
  WTOT      341B  /WALLN/                    INTEGER
  Y1       1222B                             INTEGER
  Y2       1223B                             INTEGER
```

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-----------------------VALUE

  FMAX    INTEGER                       50
  RMAX    INTEGER                       20
  WMAX    INTEGER                       75
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  DATAIN                   2      SUBROUTINE       LWALL                     0      SUBROUTINE
  ERROR                    1      SUBROUTINE       MANIP                     2      SUBROUTINE
  GETLEN    INTEGER        1      FUNCTION         PF                        5      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    100      21B                   69          450    344B                   193
    200      47B                   81          500    714B     FORMAT        199
    250     166B                  120          600  INACTIVE   DO-TERM       202
    300     246B                  157         9090 *NO REFS*                 131
    400     306B                  176         9091 *NO REFS*                 140


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  SWALLS     14B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------

  TAPE1   FMT/SEQ
  TAPE6   AUX/FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH          1231B =  665
  CM LABELLED COMMON LENGTH    2057B =  1071
  CM STORAGE USED             63000B =  26112
  COMPILE TIME                0.317 SECONDS

      1  WARNING   ERROR IN  SWALLS
```

```
      SUBROUTINE DATAIN (INSERT,LINE)                                   SWALLS   167
************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
************************************************************************COMR       3
      INTEGER RMAX                                                      COMR       4
      PARAMETER (RMAX = 20)                                             COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR       6
      INTEGER NROOMS                                                    COMR       7
      REAL ROOM                                                         COMR       8
************************************************************************COMR       9
************************************************************************COMR      10
************************************************************************COMW       1
***   COMMON FOR DATABASE OF WALL PARAMETERS                         ***COMW       2
************************************************************************COMW       3
      INTEGER WMAX                                                      COMW       4
      PARAMETER (WMAX = 75)                                             COMW       5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW       6
      COMMON /WALLC/ WALL(WMAX,4)                                       COMW       7
      INTEGER WTOT,WERR                                                 COMW       8
      REAL WDIM                                                         COMW       9
      CHARACTER *3 WALL                                                 COMW      10
* ================================                                      COMW      11
** DESCRIPTION OF ARRAYS                                                COMW      12
* ================================                                      COMW      13
*    WALL IDENTIFICATION                                                COMW      14
* --------------------------------                                      COMW      15
* DIRECTION    FROM       TO                                            COMW      16
*              ROOM       ROOM                                          COMW      17
* --------------------------------                                      COMW      18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                     COMW      19
*  A3          A3          A3                                           COMW      20
* ========================================================              COMW      21
*        WALL PARAMETERS                                                COMW      22
* --------------------------------------------------------              COMW      23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                   COMW      24
* --------------------------------------------------------              COMW      25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                         COMW      26
*  A3          F8.2        F8.2        F8.2                             COMW      27
************************************************************************COMW      28
************************************************************************COMW      29
       INTEGER  ANSWER,LOK,DOK,NOK,GETLEN,VAL, INSERT,LINE,V            SWALLS   170
       CHARACTER *3 DIR,FROM,TO,MAT                                     SWALLS   171
   99  IF ( INSERT .EQ. 1 ) THEN                                        SWALLS   172
         ANSWER = 1                                                     SWALLS   173
         INSERT = 1                                                     SWALLS   174
       ELSE                                                             SWALLS   175
100      PRINT*                                                         SWALLS   176
         PRINT*, 'IS THIS THE FIRST LAYER OF A WALL   (1) YES (2) NO'   SWALLS   177
         PRINT*, '      ENTER "0" TO ESCAPE "DATA ENTRY" MODE'          SWALLS   178
         PRINT*, ' ENTER A NUMBER!!'                                    SWALLS   179
         REWIND 1                                                       SWALLS   180
         READ(1,*,END=100) ANSWER                                       SWALLS   181
       END IF                                                           SWALLS   182
*      .....................................                            SWALLS   183
       IF (ANSWER .EQ. 0)  THEN                                         SWALLS   184
         WTOT = WTOT - 1                                                SWALLS   185
         PRINT*                                                         SWALLS   186
         PRINT*, 'DATA ENTRY MODE ABORTED'                              SWALLS   187
       END IF                                                           SWALLS   188
       IF ((ANSWER .NE. 2)                                              SWALLS   189
     +     .AND. (ANSWER .NE. 1)                                        SWALLS   190
     +     .AND. (ANSWER .NE. 0)) THEN                                  SWALLS   191
         PRINT*                                                         SWALLS   192
         PRINT*, 'INCORRECT NUMBER!!'                                   SWALLS   193
```

```
            PRINT*, ' TRY AGAIN!! -OR- ENTER "0" TO ESCAPE DATA ENTRY MODE'
            GOTO 99
          END IF
          IF (ANSWER .EQ. 1) THEN
 200        PRINT*
            PRINT*, 'ENTER DIRECTION (E. G. ''LR'')'
            REWIND 1
            READ(1,*,END=200) DIR
            IF ((DIR .NE. 'LR')
     +          .AND. (DIR .NE. 'FB')
     +          .AND. (DIR .NE. 'UD')) THEN
              PRINT*, 'DIRECTION MUST BE ''LR'' OR ''FB'' OR ''UD'''
              PRINT*, 'TRY AGAIN!!!'
              GOTO 200
            END IF
            WALL(LINE,1) = DIR
*          ..................................
  300     PRINT*
          PRINT*, 'ENTER "FROM" (E.G. ''02'' OR ''D1'')'
          REWIND 1
          READ(1,*,END=300) FROM
          LOK = 0
          DOK = 0
          NOK = 0
          IF (GETLEN(FROM) .EQ. 2) THEN
            LOK = 1
          END IF
          IF (FROM(1:1) .EQ. 'D') THEN
            V = VAL(FROM(2:2))
            IF ((V .GE. 1) .AND. (V .LE. 6)) THEN
              DOK = 1
            END IF
          END IF
          IF ((ICHAR(FROM(1:1)) .GE. 16)
     +       .AND. (ICHAR(FROM(1:1)) .LE. 25)
     +       .AND. (ICHAR(FROM(2:2)) .GE. 16)
     +       .AND. (ICHAR(FROM(2:2)) .LE. 25)
     +       .AND. (GETLEN(FROM) .EQ. 2)) THEN
            V = VAL(FROM)
            IF ((V .GE. 1) .AND. (V .LE. RMAX)) THEN
              NOK = 1
            END IF
          END IF
          IF ((LOK .EQ. 1) .AND. ((DOK .EQ. 1 ) .OR. (NOK .EQ. 1))) THEN
            WALL(LINE,2) = FROM
          ELSE
            PRINT*
            PRINT*, 'INCORRECT ENTRY.  TRY AGAIN!!'
            GOTO 300
          END IF
*         ............................
 400        PRINT*
            PRINT*, 'ENTER "TO"  (E.G. ''02'' OR ''D1'')'
            REWIND 1
            READ(1,*,END=400) TO
            LOK = 0
            DOK = 0
            NOK = 0
            IF (GETLEN(TO) .EQ. 2) THEN
              LOK = 1
            END IF
            IF (TO(1:1) .EQ. 'D') THEN
              V = VAL(TO(2:2))
              IF ((V .GE. 1) .AND. (V .LE. 6)) THEN
```

```
                DOK = 1
              END IF
            ENDIF
            IF ((ICHAR(TO(1:1)) .GE. 16)
     +         .AND. (ICHAR(TO(1:1)) .LE. 25)
     +         .AND. (ICHAR(TO(2:2)) .GE. 16)
     +         .AND. (ICHAR(TO(2:2)) .LE. 25)
     +         .AND. (GETLEN(TO) .EQ. 2)) THEN
              V = VAL (TO)
              IF ((V .GE. 1) .AND. (V .LE. RMAX)) THEN
                NOK = 1
              END IF
            ENDIF
            IF ((LOK .EQ. 1) .AND. ((DOK .EQ. 1) .OR. (NOK .EQ. 1))) THEN
              WALL (LINE,3) = TO
            ELSE
              PRINT*
              PRINT*, 'INCORRECT ENTRY.  TRY AGAIN!!'
              GOTO 400
            END IF
            IF(FROM .EQ. TO) THEN
              PRINT*
              PRINT*, 'INCORRECT ENTRY!!'
              PRINT*, '"FROM" CANNOT EQUAL "TO"'
              PRINT*, 'CHECK YOUR DATA AND REENTER "FROM" AND "TO"'
              PRINT*
              GOTO 300
            END IF
            IF ((FROM(1:1) .EQ. 'D' ) .AND. (TO(1:1) .EQ. 'D')) THEN
              PRINT*
              PRINT*,  'INCORRECT ENTRY!!'
              PRINT*, '"FROM" AND "TO" CANNOT BOTH CONTAIN "D"'
              PRINT*, '   CHECK YOUR DATA AND REENTER "FROM" AND "TO"'
              PRINT*
              GOTO 300
            END IF
*         ................................
 440        PRINT*
            PRINT*, 'ENTER HEIGHT, METERS'
            REWIND 1
            READ(1,*,END=440) WDIM(LINE,1)
*         ................................
 460        PRINT*
            PRINT*, 'ENTER WIDTH, METERS'
            REWIND 1
            READ(1,*,END=460) WDIM(LINE,2)
*         ................................
 480        PRINT*
            PRINT*, 'ENTER THICKNESS OF LAYER, CENTIMETERS'
            REWIND 1
            READ(1,*,END=480) WDIM(LINE,3)
*         ................................
 500        PRINT*
            PRINT*, 'ENTER "MATERIAL ID" (E.G. ''M01'')'
            REWIND 1
            READ(1,*,END=500) MAT
            IF ((GETLEN(MAT).EQ. 3)
     +         .AND. (MAT(1:1) .EQ. 'M')
     +         .AND. (ICHAR(MAT(2:2)) .GE. 16)
     +         .AND. (ICHAR(MAT(2:2)) .LE. 25)
     +         .AND. (ICHAR(MAT(3:3)) .GE. 16)
     +         .AND. (ICHAR(MAT(3:3)) .LE. 25)) THEN
              WALL(LINE,4) = MAT
            ELSE
```

```
              PRINT*
              PRINT*, 'INCORRECT ENTRY!!  TRY AGAIN'
              GOTO 500
            END IF
          END IF
*         ..................................
          IF (ANSWER .EQ.2) THEN
580         PRINT*
            PRINT*, 'ENTER THICKNESS OF LAYER, CENTIMETERS'
            REWIND 1
            READ(1,*,END=580) WDIM(LINE,3)
*          ...............................
600         PRINT*
            PRINT*, 'ENTER "MATERIAL ID" (E.G. ''M01'')'
            REWIND 1
            READ(1,*,END=600) MAT
            IF ((GETLEN(MAT) .EQ. 3)
     +         .AND. (MAT(1:1) .EQ. 'M')
     +         .AND. (ICHAR(MAT(2:2)) .GE. 16)
     +         .AND. (ICHAR(MAT(2:2)) .LE. 25)
     +         .AND. (ICHAR(MAT(3:3)) .GE. 16)
     +         .AND. (ICHAR(MAT(3:3)) .LE. 25)) THEN
              WALL(LINE,4) = MAT
            ELSE
              PRINT*
              PRINT*, 'INCORRECT ENTRY!!    TRY AGAIN'
              GOTO 600
            END IF
            WALL(LINE,3) = WALL(LINE-1,3)
            WALL(LINE,2) = WALL(LINE-1,2)
            WALL(LINE,1) = WALL(LINE-1,1)
            WDIM(LINE,1) = WDIM(LINE-1,1)
            WDIM(LINE,2) = WDIM(LINE-1,2)
          END IF
          RETURN
          END
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
ANSWER	1434B			INTEGER	
DIR	1441B			CHAR*3	
DOK	1436B			INTEGER	
FROM	1442B			CHAR*3	
INSERT	1	DUMMY-ARG		INTEGER	
LINE	2	DUMMY-ARG		INTEGER	
LOK	1435B			INTEGER	
MAT	1444B			CHAR*3	
NOK	1437B			INTEGER	
NROOMS	1244B	/ROOMN/		INTEGER	
RAREA	1245B	/ROOMN/		REAL	20
ROOM	0B	/ROOMN/		REAL	676
TO	1443B			CHAR*3	
V	1440B			INTEGER	
WALL	0B	/WALLC/		CHAR*3	300
WDIM	0B	/WALLN/		REAL	225
WERR	342B	/WALLN/		INTEGER	
WTOT	341B	/WALLN/		INTEGER	

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  RMAX    INTEGER                        20
  WMAX    INTEGER                        75


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN    INTEGER        1      FUNCTION
  ICHAR     INTEGER        1      INTRINSIC
  VAL       INTEGER        1      FUNCTION


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    99       7B                    43          460     477B                    171
   100      16B                    47          480     513B                    176
   200      70B                    69          500     527B                    181
   300     127B                    82          580     615B                    200
   400     264B                   116          600     631B                    205
   440     463B                   166


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DATAIN      5B     2


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES------------

  TAPE1   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH          1447B =  807
  CM LABELLED COMMON LENGTH    1766B =  1014
  CM STORAGE USED             63000B =  26112
  COMPILE TIME                0.436 SECONDS
```

```
      SUBROUTINE MANIP (QUIT,ABORT)                                     SWALLS   358
************************************************************************COMW       1
***   COMMON FOR DATABASE OF WALL PARAMETERS                         ***COMW       2
************************************************************************COMW       3
      INTEGER WMAX                                                      COMW       4
      PARAMETER (WMAX = 75)                                             COMW       5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW       6
      COMMON /WALLC/ WALL(WMAX,4)                                       COMW       7
      INTEGER WTOT,WERR                                                 COMW       8
      REAL WDIM                                                         COMW       9
      CHARACTER *3 WALL                                                 COMW      10
* ================================                                      COMW      11
** DESCRIPTION OF ARRAYS                                                COMW      12
* ================================                                      COMW      13
*    WALL IDENTIFICATION                                                COMW      14
* --------------------------------                                      COMW      15
* DIRECTION    FROM        TO                                           COMW      16
*              ROOM        ROOM                                         COMW      17
* --------------------------------                                      COMW      18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                     COMW      19
*  A3          A3           A3                                          COMW      20
* ========================================================              COMW      21
*        WALL PARAMETERS                                                COMW      22
* --------------------------------------------------------              COMW      23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                   COMW      24
* --------------------------------------------------------              COMW      25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                         COMW      26
*  A3         F8.2        F8.2        F8.2                              COMW      27
************************************************************************COMW      28
************************************************************************COMW      29
      INTEGER ABORT,ANSWER,DOK,FLAG1,LOK,N,NOK,OK,OK1,OK2,QUIT,INSERT   SWALLS   360
      INTEGER TEMP,V,X,Y,COMMAND                                        SWALLS   361
      CHARACTER * 3 DIR, FROM, TO, MAT                                  SWALLS   362
*                                                                       SWALLS   363
10    FLAG1 = 0                                                         SWALLS   364
      PRINT*                                                            SWALLS   365
      PRINT*, 'CHOOSE'                                                  SWALLS   366
      PRINT*, '   (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES'      SWALLS   367
      PRINT*, '   (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA'   SWALLS   368
      PRINT*, '   (3) DELETE LINE           (6) STORE DATA AND EXIT ',  SWALLS   369
     +        'PROGRAM'                                                 SWALLS   370
      PRINT*, '                             (7) EXIT PROGRAM WITHOUT ', SWALLS   371
     +        'STORING DATA'                                            SWALLS   372
      PRINT*, 'ENTER A NUMBER !!!'                                      SWALLS   373
      PRINT*                                                            SWALLS   374
      REWIND 1                                                          SWALLS   375
      READ(1,*,END=10) COMMAND                                          SWALLS   376
*                                                                       SWALLS   377
*---------------------------------------------------------------------- SWALLS   378
*** DISPLAY LINE ***                                                    SWALLS   379
*---------------------------------------------------------------------- SWALLS   380
      IF ( COMMAND .EQ. 1 ) THEN                                        SWALLS   381
*                                                                       SWALLS   382
*** INDICATE EMPTY DATA FILE                                            SWALLS   383
        IF ( WTOT .EQ. 0 ) THEN                                         SWALLS   384
          PRINT*                                                        SWALLS   385
          PRINT*, 'DATA FILE IS EMPTY !!!'                              SWALLS   386
*                                                                       SWALLS   387
*** ENTER NUMBER OF LINE TO BE DISPLAYED                                SWALLS   388
        ELSE                                                            SWALLS   389
100       PRINT*                                                        SWALLS   390
          PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED'      SWALLS   391
          PRINT*, '     ( ENTER "0" TO ESCAPE DISPLAY MODE )'           SWALLS   392
          REWIND 1                                                      SWALLS   393
```

```
          READ(1,*,END=100) N
*
*** CHECK VALIDITY OF LINE NUMBER
          IF ( (N .GT. WTOT) .OR. (N .LT. 0) ) THEN
            PRINT*
            PRINT*, 'INCORRECT NUMBER !!!!!!        TRY AGAIN !!!'
            PRINT*, '             -OR-  ENTER "0" TO ESCAPE FROM ',
     +              '"DISPLAY" MODE'
            GOTO 100
*
*** ABORT 'DISPLAY' MODE
          ELSE IF ( N .EQ. 0 ) THEN
            PRINT*
            PRINT*, '    "DISPLAY" MODE ABORTED !!!'
*
*** DISPLAY LINE OF DATA
          ELSE IF ( (N .GT. 0) .AND. (N .LE. WTOT) ) THEN
            PRINT*
            CALL DISPLAY( N, COMMAND)
*
          END IF
        END IF
      END IF
*
*-----------------------------------------------------------------------
*** INSERT LINE ***
*-----------------------------------------------------------------------
      IF ( COMMAND .EQ. 2 ) THEN
***************************************************************
*   CHECK TO SEE IF THERE IS ENOUGH ARRAY SPACE
***************************************************************
      IF(WTOT.GE.WMAX) THEN
      WTOT = WMAX
      PRINT *, 'DATA ENTRY ABORTED.'
      PRINT *, 'MAXIMUM NUMBER OF DATA LINES IN FILE WOULD'
      PRINT *, ' HAVE BEEN EXCEEDED. NO MORE THAN ',WMAX
      PRINT *, ' DATA LINES ARE ALLOWED.'
      PRINT *, ' TO INCREASE THE MAXIMUM NUMBER OF ENTRIES ALLOWED,'
      PRINT *, ' CHANGE THE PARAMETER "WMAX" IN EACH COMMON OF'
      PRINT *, ' EVERY SUBROUTINE (THERE ARE FOUR PLACES).'
      PRINT *, ' THEN RECOMPILE THE PROGRAM.'
      GOTO 10
      ENDIF
***************************************************************
*
*** INDICATE EMPTY DATA FILE
        IF ( WTOT .EQ. 0 ) THEN
          PRINT*
          PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** REQUEST NUMBER OF LINE BEFORE WHICH INSERTION IS TO BE MADE
        ELSE
200       PRINT*
          PRINT*, 'SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS ',
     +            'TO BE INSERTED'
          PRINT*, '     ( ENTER "0" TO ESCAPE "INSERTION" MODE )'
          REWIND 1
          READ(1,*,END=200) N
*
*** CHECK FOR VALID LINE NUMBER
          IF ( ( N .LT. 0 ) .OR. ( N .GT. WTOT ) ) THEN
            PRINT*
            PRINT*, 'INCORRECT NUMBER !!!'
            PRINT*, '      TRY AGAIN !!!  -OR-  ENTER "0" TO ESCAPE',
```

```
     +              '"INSERTION" MODE'
            GOTO 200
*
*** ABORT INSERTION MODE
          ELSE IF ( N .EQ. 0 ) THEN
            PRINT*
            PRINT*, '       "INSERTION" MODE ABORTED'
*
*** MAKE ROOM FOR NEW LINE OF DATA
          ELSE IF ( (N .GT. 0) .AND. (N .LE. WTOT) ) THEN
            DO 230 X = WTOT,N,-1
              DO 210 Y = 1,4
                WALL(X+1,Y) = WALL(X,Y)
 210          CONTINUE
              DO 220 Y = 1,3
                WDIM(X+1,Y) = WDIM(X,Y)
 220          CONTINUE
 230        CONTINUE
*
*** ENTER DATA FOR NEW LINE
            WTOT = WTOT + 1
            CALL DATAIN (1,N)
*
*** INITIALIZE FLAGS
            OK1 = 0
            OK2 = 0
            OK = 0
*
*** TEST VALIDITY OF DATA
*
***TEST IF NEW LAYER BELONGS TO THE NEXT WALL
            IF ( ( WALL(N,1) .EQ. WALL(N+1,1) )
     +           .AND. ( WALL(N,2) .EQ. WALL(N+1,2) )
     +           .AND. ( WALL(N,3) .EQ. WALL(N+1,3) ) ) THEN
              IF ( ( WDIM(N,1) .EQ. WDIM(N+1,1) )
     +             .AND. ( WDIM(N,2) .EQ. WDIM(N+1,2) ) ) THEN
                OK1 = 1
              END IF
            END IF
*
*** TEST IF NEW LAYER BELONGS TO PREVIOUS WALL
            IF ( N .GT. 1 ) THEN
              IF ( ( WALL(N,1) .EQ. WALL(N-1,1) )
     +             .AND. ( WALL(N,2) .EQ. WALL(N-1,2) )
     +             .AND. ( WALL(N,3) .EQ. WALL(N-1,3) ) ) THEN
                IF ( ( WDIM(N,1) .EQ. WDIM(N-1,1) )
     +               .AND. ( WDIM(N,2) .EQ. WDIM(N-1,2) ) ) THEN
                  OK2 = 1
                END IF
              END IF
            END IF
*
            IF ( ( OK1 .EQ. 1 ) .OR. ( OK2 .EQ. 1 ) ) THEN
              OK = 1
            END IF
*
            IF ( OK .EQ. 1 ) THEN
              PRINT*
              PRINT*, 'THE FOLLOWING LINE HAS BEEN ADDED AS LINE ', N
              CALL DISPLAY( N, COMMAND)
*
*** REJECT DATA IF DATA DOESN'T MATCH PREVIOUS OR NEXT LAYER
            ELSE IF ( OK .EQ. 0 ) THEN
              PRINT*
```

```
               PRINT*, 'YOUR DATA WAS NOT ACCEPTED !!!'                  SWALLS
               PRINT*, '  YOUR DATA MUST REPRESENT A LAYER ',            SWALLS
     +                  'IN AN EXISTING WALL'                            SWALLS
               PRINT*, '  I.E.  THE DIRECTION, FROM, TO, HEIGHT, AND ',  SWALLS
     +                'WIDTH'                                            SWALLS
               PRINT*, '        PARAMETERS MUST MATCH THE WALL JUST ',   SWALLS
     +                  'BEFORE'                                         SWALLS
               PRINT*, '        OR JUST AFTER YOUR SPECIFIED INSERTION ',SWALLS
     +                  'POINT'                                          SWALLS
               PRINT*                                                    SWALLS
               PRINT*, 'THE FOLLOWING DISPLAYS'                          SWALLS
               IF ( N .GT. 1 ) PRINT*,'THE LINE BEFORE YOUR LINE,'       SWALLS
               PRINT*, 'YOUR LINE, AND THE LINE AFTER'                   SWALLS
               PRINT*                                                    SWALLS
*                                                                        SWALLS
*** DISPLAY LINES OF DATA                                                SWALLS
               IF ( N .GT. 1 ) CALL DISPLAY ( N-1, COMMAND )             SWALLS
               CALL DISPLAY( N, COMMAND)                                 SWALLS
               CALL DISPLAY ( N+1, COMMAND)                              SWALLS
*                                                                        SWALLS
*** REMOVE THE LINE OF INCORRECTLY ENTERED DATA                          SWALLS
               DO 270  X = N,WTOT                                        SWALLS
                 DO 250  Y = 1,4                                         SWALLS
                   WALL(X,Y) = WALL(X+1,Y)                               SWALLS
 250             CONTINUE                                                SWALLS
                 DO 260  Y = 1,3                                         SWALLS
                   WDIM(X,Y) = WDIM(X+1,Y)                               SWALLS
 260             CONTINUE                                                SWALLS
 270           CONTINUE                                                  SWALLS
               WTOT = WTOT - 1                                           SWALLS
            END IF                                                       SWALLS
          END IF                                                         SWALLS
          END IF                                                         SWALLS
          END IF                                                         SWALLS
*                                                                        SWALLS
*----------------------------------------------------------------------  SWALLS
*** DELETE LINE ***                                                      SWALLS
*----------------------------------------------------------------------  SWALLS
          IF ( COMMAND .EQ. 3 ) THEN                                     SWALLS
*                                                                        SWALLS
*** INDICATE EMPTY DATA FILE                                             SWALLS
            IF ( WTOT .EQ. 0 ) THEN                                      SWALLS
              PRINT*                                                     SWALLS
              PRINT*, 'DATA FILE IS EMPTY !!!'                           SWALLS
*                                                                        SWALLS
*** READ NUMBER OF LINE TO BE DELETED                                    SWALLS
            ELSE                                                         SWALLS
 300          PRINT*                                                     SWALLS
              PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DELETED'     SWALLS
              PRINT*, '   (ENTER "0" TO ESCAPE DELETION MODE)'           SWALLS
              REWIND 1                                                   SWALLS
              READ(1,*,END=300) N                                        SWALLS
*                                                                        SWALLS
*** CHECK VALIDITY OF LINE NUMBER                                        SWALLS
              IF ( (N .GT. WTOT ) .OR. ( N .LT. 0 ) ) THEN               SWALLS
                PRINT*                                                   SWALLS
                PRINT*, ' INCORRECT NUMBER !!!'                          SWALLS
                PRINT*, '    TRY AGAIN !!!  -OR-  ENTER "0" TO ESCAPE FROM',SWALLS
     +                  '"DELETE" MODE'                                  SWALLS
                GOTO 300                                                 SWALLS
*                                                                        SWALLS
*** ABORT 'DELETE' MODE                                                  SWALLS
              ELSE IF ( N .EQ. 0 ) THEN                                  SWALLS
                PRINT*, '  "DELETE" MODE ABORTED'                        SWALLS
```

```
*
*** DOUBLE CHECK CHOICE OF LINE TO BE DELETED
            ELSE IF (( N .GT. 0 ) .AND. ( N .LE. WTOT )) THEN
              PRINT*
              PRINT*, 'DOUBLE CHECK !!!'
              PRINT*, '     DO YOU WANT TO DELETE THE FOLLOWING LINE?:'
              CALL DISPLAY( N, COMMAND)
 305          PRINT*, '   ENTER (1) YES  OR   (2) NO'
              REWIND 1
              READ(1,*,END=305) ANSWER
*
*** DELETE LINE
              IF ( ANSWER .EQ. 1 ) THEN
                DO 330 X = N, WTOT - 1
                  DO 310 Y = 1,4
                    WALL(X,Y) = WALL(X+1,Y)
 310              CONTINUE
                  DO 320 Y = 1,3
                    WDIM(X,Y) = WDIM(X+1,Y)
 320              CONTINUE
 330            CONTINUE
                WTOT = WTOT - 1
                PRINT*
                PRINT*, 'LINE # ',N,' DELETED'
              END IF
*
            END IF
          END IF
        END IF
*
*-----------------------------------------------------------------------
*** DISPLAY ALL DATA ***
*-----------------------------------------------------------------------
        IF ( COMMAND .EQ. 4 ) THEN
*
*** INDICATE EMPTY DATA FILE
          IF ( WTOT .EQ. 0 ) THEN
            PRINT*
            PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** DISPLAY DATA
          ELSE
            PRINT*
            CALL DISPLAY( N, COMMAND)
*
          END IF
        END IF
*
*-----------------------------------------------------------------------
*** ADD DATA ***
*-----------------------------------------------------------------------
        IF ( COMMAND .EQ. 5 ) THEN
*
*** ENTER DATA
*****************************************************************
*   CHECK TO SEE IF THERE IS ENOUGH ARRAY SPACE
*****************************************************************
 500    IF(WTOT.GE.WMAX) THEN
        WTOT = WMAX
        PRINT *, 'DATA ENTRY ABORTED.'
        PRINT *, 'MAXIMUM NUMBER OF DATA LINES IN FILE WOULD'
        PRINT *, ' HAVE BEEN EXCEEDED. NO MORE THAN ',WMAX
        PRINT *, ' DATA LINES ARE ALLOWED.'
        PRINT *, ' TO INCREASE THE MAXIMUM NUMBER OF ENTRIES ALLOWED,'
```

```
      PRINT *, ' CHANGE THE PARAMETER "WMAX" IN EACH COMMON OF'
      PRINT *, ' EVERY SUBROUTINE (THERE ARE FOUR PLACES).'
      PRINT *, ' THEN RECOMPILE THE PROGRAM.'
      GOTO 10
      ENDIF
***************************************************************
        WTOT = WTOT + 1
        CALL DATAIN (0,WTOT)
510     PRINT*
        PRINT*, 'DO YOU WANT TO ENTER MORE DATA?   (1) YES  (2) NO'
        PRINT*, '   ENTER A NUMBER !!!'
        REWIND 1
        READ(1,*,END=510) ANSWER
*
*** CHECK VALIDITY OF NUMBER
        IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN
          GOTO 510
*
*** ENTER MORE DATA
        ELSE IF ( ANSWER .EQ. 1 ) THEN
          GOTO 500
*
*** DISCONTINUE DATA ENTRY
        ELSE IF ( ANSWER .EQ. 2 ) THEN
          PRINT*
          PRINT*, 'DATA ENTRY DISCONTINUED'
*
        END IF
      END IF
*
*-----------------------------------------------------------------------
*** STORE DATA AND PROGRAM ***
*-----------------------------------------------------------------------
      IF ( COMMAND .EQ. 6 ) THEN
 600    PRINT*
        PRINT*, 'DOUBLE CHECK !!!'
        PRINT*, '    DO YOU YOU WANT TO STORE THIS DATA AND END PROG'
        PRINT*, '  NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE '
        PRINT*, '  OF THE SAME NAME !!!'
        PRINT*, '         ENTER A NUMBER: (1) YES   (2) NO'
        REWIND 1
        READ(1,*,END=600) ANSWER
*
*** SET FLAG FOR STORING DATA IN THE MAIN PROGRAM
        IF ( ANSWER .EQ. 1 ) THEN
          QUIT = 1
          RETURN
*
*** ABORT 'STORING' MODE
        ELSE IF ( ANSWER .EQ. 2 ) THEN
          PRINT*
          PRINT*, '    "STORING" MODE DISCONTINUED'
*
*** CHECK VALIDITY OF ANSWER
        ELSE IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN
          GOTO 600
*
        ENDIF
      ENDIF
*
*-----------------------------------------------------------------------
*** END PROGRAM WITHOUT STORING DATA ***
*-----------------------------------------------------------------------
      IF ( COMMAND .EQ. 7 ) THEN
```

```
  700   PRINT*
        PRINT*, 'DOUBLE CHECK !!!'
        PRINT*, '    DO YOU WANT TO END THIS PROGRAM ',
     +        'WITHOUT STORING DATA?'
        PRINT*, '          ENTER A NUMBER: (1) YES   (2) NO'
        REWIND 1
        READ(1,*,END=700) ANSWER
*
*** SET FLAG FOR ABORTING PROGRAM IN THE MAIN PROGRAM
        IF ( ANSWER .EQ. 1 ) THEN
          ABORT = 1
          RETURN
*
*** ABORT 'STORING' MODE
        ELSE IF ( ANSWER .EQ. 2 ) THEN
          PRINT*
          PRINT*, '    "ABORTION" MODE DISCONTINUED'
*
*** CHECK VALIDITY OF ANSWER
        ELSE IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN
          GOTO 700
*
        ENDIF
      ENDIF
*
*----------------------------------------------------------------------
*** LOOP TO BEGINNING OF 'MANIP' SUBROUTINE
*----------------------------------------------------------------------
      GOTO 10
*
      END
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
ABORT	2	DUMMY-ARG		INTEGER	
ANSWER	2227B			INTEGER	
COMMAND	2237B			INTEGER	
DIR	NONE		UNUSED/*S*	CHAR*3	
DOK	NONE		UNUSED/*S*	INTEGER	
FLAG1	2230B			INTEGER	
FROM	NONE		UNUSED/*S*	CHAR*3	
INSERT	NONE		UNUSED/*S*	INTEGER	
LOK	NONE		UNUSED/*S*	INTEGER	
MAT	NONE		UNUSED/*S*	CHAR*3	
N	2231B			INTEGER	
NOK	NONE		UNUSED/*S*	INTEGER	
OK	2232B			INTEGER	
OK1	2233B			INTEGER	
OK2	2234B			INTEGER	
QUIT	1	DUMMY-ARG		INTEGER	
TEMP	NONE		UNUSED/*S*	INTEGER	
TO	NONE		UNUSED/*S*	CHAR*3	
V	NONE		UNUSED/*S*	INTEGER	
WALL	0B	/WALLC/		CHAR*3	300
WDIM	0B	/WALLN/		REAL	225
WERR	342B	/WALLN/		INTEGER	
WTOT	341B	/WALLN/		INTEGER	
X	2235B			INTEGER	
Y	2236B			INTEGER	

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  WMAX     INTEGER                        75


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  DATAIN                    2      SUBROUTINE
  DISPLAY                   2      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF

     10       7B                    35          300     611B                   240
    100      50B                    61          305     660B                   264
    200     161B                   117          310  INACTIVE   DO-TERM        273
    210  INACTIVE   DO-TERM        142          320  INACTIVE   DO-TERM        276
    220  INACTIVE   DO-TERM        145          330  INACTIVE   DO-TERM        277
    230  INACTIVE   DO-TERM        146          500    1003B                   314
    250  INACTIVE   DO-TERM        217          510    1035B                   329
    260  INACTIVE   DO-TERM        220          600    1076B                   355
    270  INACTIVE   DO-TERM        221          700    1150B                   385


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  MANIP       5B      2


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH           2253B =  1195
  CM LABELLED COMMON LENGTH      475B =  317
  CM STORAGE USED              65000B =  27136
  COMPILE TIME                 0.625 SECONDS
```

```
      SUBROUTINE DISPLAY (LINE, COMMAND)                                SWALLS   745
************************************************************************COMW       1
***   COMMON FOR DATABASE OF WALL PARAMETERS                         ***COMW       2
************************************************************************COMW       3
      INTEGER WMAX                                                      COMW       4
      PARAMETER (WMAX = 75)                                             COMW       5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW       6
      COMMON /WALLC/ WALL(WMAX,4)                                       COMW       7
      INTEGER WTOT,WERR                                                 COMW       8
      REAL WDIM                                                         COMW       9
      CHARACTER *3 WALL                                                 COMW      10
* =================================                                     COMW      11
** DESCRIPTION OF ARRAYS                                                COMW      12
* =================================                                     COMW      13
*    WALL IDENTIFICATION                                                COMW      14
* ---------------------------------                                     COMW      15
* DIRECTION    FROM       TO                                            COMW      16
*              ROOM       ROOM                                          COMW      17
* ---------------------------------                                     COMW      18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                     COMW      19
*  A3          A3          A3                                           COMW      20
* =========================================================             COMW      21
*        WALL PARAMETERS                                                COMW      22
* ---------------------------------------------------------             COMW      23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                   COMW      24
* ---------------------------------------------------------             COMW      25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                         COMW      26
*  A3          F8.2        F8.2        F8.2                             COMW      27
************************************************************************COMW      28
************************************************************************COMW      29
      INTEGER LINE, COMMAND, N                                          SWALLS   747
1000  FORMAT (8(3X,A))                                                  SWALLS   748
2000  FORMAT (4X,I3,8X,A3,7X,A3,3X,A3,2X,F6.2,2X,F6.2,5X,F6.2,7X,A3)    SWALLS   749
      PRINT 1000, 'LINE #','DIRECTION','FROM','TO','HEIGHT',            SWALLS   750
     +            'WIDTH','THICKNESS','MATERIAL'                        SWALLS   751
      IF ( COMMAND .EQ  4 ) THEN                                        SWALLS   752
        DO 10 N = 1,WTOT                                                SWALLS   753
          PRINT 2000, N,WALL(N,1),WALL(N,2),WALL(N,3),WDIM(N,1),        SWALLS   754
     +                WDIM(N,2),WDIM(N,3),WALL(N,4)                     SWALLS   755
10      CONTINUE                                                        SWALLS   756
      ELSE                                                              SWALLS   757
          PRINT 2000, LINE,WALL(LINE,1),WALL(LINE,2),WALL(LINE,3),      SWALLS   758
     +         WDIM(LINE,1),WDIM(LINE,2),WDIM(LINE,3), WALL(LINE,4)     SWALLS   759
      END IF                                                            SWALLS   760
      RETURN                                                            SWALLS   761
      END                                                               SWALLS   762
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  COMMAND      2  DUMMY-ARG                  INTEGER
  LINE         1  DUMMY-ARG                  INTEGER
  N         244B                             INTEGER
  WALL        0B  /WALLC/                    CHAR*3          300
  WDIM        0B  /WALLN/                    REAL            225
  WERR      342B  /WALLN/                    INTEGER
  WTOT      341B  /WALLN/                    INTEGER
```

```
SUBROUTINE DISPLAY  74/175  OPT=0

--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  WMAX    INTEGER                          75


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        40
  1000    150B     FORMAT         32
  2000    152B     FORMAT         33


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DISPLAY     5B     2


--STATISTICS--

  PROGRAM-UNIT LENGTH            250B =  168
  CM LABELLED COMMON LENGTH      475B =  317
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.085 SECONDS
```

```
     1        INTEGER FUNCTION VAL(STRING)                     SWALLS     763
     2 C** RETURNS THE INTEGER VALUE OF A STRING.              SWALLS     764
     3        INTEGER NUMBER, X,L,EXP,DIGIT,GETLEN             SWALLS     765
     4        CHARACTER * (*) STRING                           SWALLS     766
     5        L = GETLEN(STRING)                               SWALLS     767
     6        NUMBER = 0                                       SWALLS     768
     7        DO 10 X = L,1,-1                                 SWALLS     769
     8        EXP = L - X                                      SWALLS     770
     9        DIGIT = ICHAR(STRING(X:X)) - 16                  SWALLS     771
    10        NUMBER = NUMBER + DIGIT*10**EXP                  SWALLS     772
    11 10     CONTINUE                                         SWALLS     773
    12        VAL = NUMBER                                     SWALLS     774
    13        RETURN                                           SWALLS     775
    14        END                                              SWALLS     776
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  DIGIT      76B                             INTEGER
  EXP        75B                             INTEGER
  L          74B                             INTEGER
  NUMBER     72B                             INTEGER
  STRING      1   DUMMY-ARG                  CHAR*(*)
  VAL        71B                             INTEGER
  X          73B                             INTEGER


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN    INTEGER        1      FUNCTION
  ICHAR     INTEGER        1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        11


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  VAL         6B     1


--STATISTICS--

  PROGRAM-UNIT LENGTH            102B =  66
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.039 SECONDS
```

```
      INTEGER FUNCTION GETLEN (STRING)
C
C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING
C
C
C ARGUMENT DEFINITIONS --
C   READ ARGUMENTS
C     STRING - STRING WHOSE LENGTH IS TO BE DETERMINED
C
      CHARACTER * (*) STRING
C
C FUNCTION PARAMETERS
      CHARACTER * 1 BLANK
      PARAMETER (BLANK = ' ')
C
C LOCAL VARIABLES
      INTEGER NEXT
C
C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK
      DO 10 NEXT = LEN(STRING),1,-1
         IF (STRING(NEXT : NEXT) .NE. BLANK) THEN
            GETLEN = NEXT
            RETURN
         ENDIF
   10 CONTINUE
C
C ALL CHARACTERS ARE BLANKS
      GETLEN = 0
C
      RETURN
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  GETLEN     63B                            INTEGER
  NEXT       64B                            INTEGER
  STRING      1   DUMMY-ARG                 CHAR*(*)


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-----------------------VALUE

  BLANK   CHAR*1                       ' '


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  LEN       INTEGER        1       INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        25
```

```
--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 GETLEN      6B      1


--STATISTICS--

 PROGRAM-UNIT LENGTH              70B =  56
 CM STORAGE USED               61000B =  25088
 COMPILE TIME                  0.037 SECONDS
```

```
      SUBROUTINE LWALL                                                  LWALL     1
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LWALL     2
*!!!                                                                 !!!LWALL     3
*!!!  LOAD THE CONTENTS OF THE FILE 'WALLS' INTO ARRAYS WALL AND WDIM.  LWALL     4
*!!!                                                                 !!!LWALL     5
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LWALL     6
************************************************************************LWALL     7
************************************************************************COMW      1
***   COMMON FOR DATABASE OF WALL PARAMETERS                         ***COMW      2
************************************************************************COMW      3
      INTEGER WMAX                                                      COMW      4
      PARAMETER (WMAX = 75)                                             COMW      5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW      6
      COMMON /WALLC/ WALL(WMAX,4)                                       COMW      7
      INTEGER WTOT,WERR                                                 COMW      8
      REAL WDIM                                                         COMW      9
      CHARACTER *3 WALL                                                 COMW     10
* ==================================                                    COMW     11
** DESCRIPTION OF ARRAYS                                                COMW     12
* ==================================                                    COMW     13
*    WALL IDENTIFICATION                                                COMW     14
* ----------------------------------                                    COMW     15
* DIRECTION    FROM       TO                                            COMW     16
*              ROOM       ROOM                                          COMW     17
* ----------------------------------                                    COMW     18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                     COMW     19
*  A3          A3          A3                                           COMW     20
* ============================================================          COMW     21
*        WALL PARAMETERS                                                COMW     22
* ------------------------------------------------------------          COMW     23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                   COMW     24
* ------------------------------------------------------------          COMW     25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                         COMW     26
*  A3         F8.2        F8.2        F8.2                              COMW     27
************************************************************************COMW     28
************************************************************************COMW     29
************************************************************************COMF      1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF      2
************************************************************************COMF      3
      INTEGER FMAX                                                      COMF      4
      PARAMETER (FMAX = 50)                                             COMF      5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF      6
     $                 FTOT                                             COMF      7
      COMMON /INITILC/ BLDG                                             COMF      8
      CHARACTER * 5 BLDG                                                COMF      9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF     10
      INTEGER QUALITY, FERR, FTOT                                       COMF     11
************************************************************************COMF     12
************************************************************************COMF     13
*****************************************************                   LWALL    10
*  DECLARATION OF VARIABLES                                             LWALL    11
*****************************************************                   LWALL    12
      INTEGER GETLEN, R, C                                              LWALL    13
      CHARACTER * 7 NAME, PFN                                           LWALL    14
*****************************************************                   LWALL    15
*                                                                       LWALL    16
*****************************************************                   LWALL    17
      NAME = 'B'//BLDG(1:GETLEN(BLDG))//'W'                             LWALL    18
      PFN = NAME (1:GETLEN(NAME))                                       LWALL    19
      WERR = 0                                                          LWALL    20
      CALL PF  ('GET',0,PFN(1:GETLEN(PFN)),'RC',WERR)                   LWALL    21
      IF ( WERR .EQ. 0 ) THEN                                           LWALL    22
        OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',                       LWALL    23
     $        STATUS='OLD', ACCESS='SEQUENTIAL')                        LWALL    24
```

```
     65 1000     FORMAT (1X,4(1X,A3),3(1X,F8.2))                         LWALL      25
     66            WTOT = 0                                                LWALL      26
     67            DO 10 R = 1,WMAX                                        LWALL      27
     68              READ (3,1000,END=20)(WALL(R,C),C=1,4),(WDIM(R,C),C=1,3) LWALL    28
     69              WTOT = WTOT + 1                                       LWALL      29
     70 10         CONTINUE                                                LWALL      30
     71 20         CONTINUE                                                LWALL      31
     72            CLOSE(3,STATUS='DELETE')                                LWALL      32
     73          ELSE IF ( WERR .EQ. 2 ) THEN                              LWALL      33
     74            CALL WARNING (7)                                        LWALL      34
     75          ELSE                                                      LWALL      35
     76            CALL WARNING (8)                                        LWALL      36
     77          END IF                                                    LWALL      37
     78          RETURN                                                    LWALL      38
     79          END                                                       LWALL      39
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG       2B  /INITILN/                  REAL
  BLDG        0B  /INITILC/                  CHAR*5
  C         255B                             INTEGER
  FERR       66B  /INITILN/                  INTEGER
  FREQ        0B  /INITILN/                  REAL
  FREQA       4B  /INITILN/                  REAL           50
  FTOT       67B  /INITILN/                  INTEGER
  NAME      256B                             CHAR*7
  PFN       257B                             CHAR*7
  QUALITY     1B  /INITILN/                  INTEGER
  R         254B                             INTEGER
  RFLAG       3B  /INITILN/                  REAL
  WALL        0B  /WALLC/                    CHAR*3        300
  WDIM        0B  /WALLN/                    REAL          225
  WERR      342B  /WALLN/                    INTEGER
  WTOT      341B  /WALLN/                    INTEGER
```

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  FMAX    INTEGER                        50
  WMAX    INTEGER                        75
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN    INTEGER         1      FUNCTION
  PF                        5      SUBROUTINE
  WARNING                   1      SUBROUTINE
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM         70
     20     117B                    71
   1000     155B    FORMAT          65
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  LWALL       5B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE3   AUX/FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH              265B =  181
  CM LABELLED COMMON LENGTH        566B =  374
  CM STORAGE USED                63000B =  26112
  COMPILE TIME                  0.105 SECONDS
```

```
 1        SUBROUTINE WARNING(ERR)                                       WARNING      1
 2        INTEGER ERR, ERRM                                             WARNING      2
 3        CHARACTER*45 MESSAGE(20)                                      WARNING      3
 4        DATA MESSAGE( 1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING  4
 5        DATA MESSAGE( 2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE    '/ WARNING  5
 6        DATA MESSAGE( 3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG   '/ WARNING  6
 7        DATA MESSAGE( 4)/'FILE HANDLING PROBLEM ON "MATTER FILE        '/ WARNING  7
 8        DATA MESSAGE( 5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING  8
 9        DATA MESSAGE( 6)/'FILE HANDLING PROBLEM ON "TYPE" FILE         '/ WARNING  9
10        DATA MESSAGE( 7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING 10
11        DATA MESSAGE( 8)/'FILE HANDLING PROBLEM ON "WALL" FILE         '/ WARNING 11
12        DATA MESSAGE( 9)/'HEIGHT AND WIDTH OF ROOM MISSING             '/ WARNING 12
13        DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING                    '/ WARNING 13
14        DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG       '/ WARNING 14
15        DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE         '/ WARNING 15
16        DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING 16
17        DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING 17
18        DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING 18
19        DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING 19
20        DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING 20
21        DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING 21
22        DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING 22
23        DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING 23
24        ERRM=12                                                       WARNING     24
25        IERR = ERR                                                    WARNING     25
26        IF(ERR.GT.ERRM) IERR=20                                       WARNING     26
27        WRITE(6,20)                                                   WARNING     27
28        WRITE(6,10) ERR,MESSAGE(IERR)                                 WARNING     28
29        WRITE(6,20)                                                   WARNING     29
30 10     FORMAT(' ***WARNING NUMBER = ',I5,' ***  ',A45)               WARNING     30
31 20     FORMAT('     ')                                               WARNING     31
32        RETURN                                                        WARNING     32
33        END                                                           WARNING     33
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ERR          1   DUMMY-ARG                  INTEGER
  ERRM       60B                              INTEGER
  IERR      213B                              INTEGER
  MESSAGE    61B                              CHAR*45         20


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10      34B     FORMAT          30
    20      42B     FORMAT          31


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  WARNING     5B      1
```

```
FTN 5.1+552          83/12/24. 11.29.46 PAGE     28
SUBROUTINE WARNING  74/175   OPT=0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE6    FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH            216B =   142
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.060 SECONDS
```

```
      SUBROUTINE ERROR(IERR)
      CHARACTER*45 MESSAGE(20)
      DATA MESSAGE( 1)/'MATERIALS DATA BASE IS EMPTY                '/
      DATA MESSAGE( 2)/'FREQUENCY IS OUT OF RANGE                   '/
      DATA MESSAGE( 3)/'THIS MATERIAL IS NOT IN DATA BASE           '/
      DATA MESSAGE( 4)/'DENOMINATOR IS ZERO                         '/
      DATA MESSAGE( 5)/'FILE HANDLING ERROR                         '/
      DATA MESSAGE( 6)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE( 7)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE( 8)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE( 9)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(10)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(11)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(12)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(13)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(14)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(15)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(16)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(17)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(18)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(19)/'ERROR CODE IS OUT OF RANGE                  '/
      DATA MESSAGE(20)/'ERROR CODE IS OUT OF RANGE                  '/
      IERRM=5
      IF(IERR.GT.IERRM) IERR=20
      WRITE(6,10) IERR,MESSAGE(IERR)
   10 FORMAT(' ***ERROR NUMBER = ',I5,' ***  ',A45)
      CALL PMDSTOP
      STOP 'ERROR'
      END
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 IERR        1   DUMMY-ARG                  INTEGER
 IERRM     210B                             INTEGER
 MESSAGE    56B                             CHAR*45          20


--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 PMDSTOP                  0      SUBROUTINE


--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

   10     36B     FORMAT          26


--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 ERROR       5B     1
```

FTN 5.1+552 83/12/24. 11.29.46 PAGE 30
SUBROUTINE ERROR 74/175 OPT=0

```
--I/O UNITS--(LO=A)
-NAME--- PROPERTIES-------------

 TAPE6   FMT/SEQ


--STATISTICS--

 PROGRAM-UNIT LENGTH            213B =  139
 CM STORAGE USED              61000B =  25088
 COMPILE TIME                 0.056 SECONDS
```

Appendix 9.5 Listing of Computer Program STYPES

```
      PROGRAM STYPES (INPUT,TAPE1=INPUT)                                STYPES     1
*                                                                       STYPES     2
*THIS INTERACTIVE PROGRAM INPUTS THE DATA DESCRIBING EACH TYPE          STYPES     3
*IN THE BUILDING AND STORES IT.  THE FILE NAME IS CREATED BY            STYPES     4
*ATTACHING "B" TO THE FRONT OF AND "W" TO THE BACK OF THE BUILDING      STYPES     5
*IDENTIFICATION. THE BUILDING IDENTIFICATION CAN BE NO MORE             STYPES     6
*THAN 5 ALPHANUMERIC CHARACTERS.                                        STYPES     7
                                                                        STYPES     8
************************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF       2
************************************************************************COMF       3
      INTEGER FMAX                                                      COMF       4
      PARAMETER (FMAX = 50)                                             COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF       6
     $                 FTOT                                             COMF       7
      COMMON /INITILC/ BLDG                                             COMF       8
      CHARACTER * 5 BLDG                                                COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF      10
      INTEGER QUALITY, FERR, FTOT                                       COMF      11
************************************************************************COMF      12
************************************************************************COMF      13
************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
************************************************************************COMR       3
      INTEGER RMAX                                                      COMR       4
      PARAMETER (RMAX = 20)                                             COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR       6
      INTEGER NROOMS                                                    COMR       7
      REAL ROOM                                                         COMR       8
************************************************************************COMR       9
************************************************************************COMR      10
************************************************************************COMT       1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT       2
************************************************************************COMT       3
      INTEGER TMAX                                                      COMT       4
      PARAMETER (TMAX=35)                                               COMT       5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT       6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT       7
      INTEGER TTOT,TDBTOT,TERR                                          COMT       8
      REAL TDIM,TDB2                                                    COMT       9
      CHARACTER * 3 TYPE,TDB1                                           COMT      10
*=================================================                      COMT      11
* DESCRIPTION OF ARRAYS                                                 COMT      12
*=================================================                      COMT      13
*   ID        MATERIAL    FRAME MATERIAL                                COMT      14
*-------------------------------------------------                      COMT      15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                      COMT      16
* A3          A3          A3                                            COMT      17
*=================================================                      COMT      18
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT      19
*                         THICKNESS   ABOVE FLOOR                       COMT      20
*-------------------------------------------------                      COMT      21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT      22
*  F8.2        F8.2        F8.2        F8.2                             COMT      23
*=================================================                      COMT      24
*  ID       ATTENUATION    AREA                                         COMT      25
* ------------------------------------------------                      COMT      26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                     COMT      27
*   A3          E9.3         E9.3                                       COMT      28
************************************************************************COMT      29
************************************************************************COMT      30
      INTEGER GETLEN,QUIT,ABORT,ANSWER,OLDFILE,N,Y1,Y2,LINE             STYPES    12
      INTEGER IERR                                                      STYPES    13
      CHARACTER * 7  PFN                                                STYPES    14
```

```
*
* INITIALIZATION
      QUIT = 0
      TTOT = 0
      ABORT = 0
100   PRINT*
      PRINT *, 'ENTER BUILDING IDENTIFICATION  (E.G. ''101'')'
      PRINT *, '    (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)'
      REWIND 1
      READ(1,*,END=100) BLDG

      IF ( GETLEN(BLDG) .GT. 5 ) THEN
        GO TO 100
      END IF
      PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'T'
*
*** LOAD DATA ID EXISTING FILE IF NECESSARY
200   PRINT*
      PRINT*,'WILL THIS BE'
      PRINT*,'   (1) A MODIFICATION OF AN EXISTING FILE?'
      PRINT*,'   (2) A NEW FILE?'
      PRINT*,'ENTER A NUMBER  !!!'
      REWIND 1
      READ(1,*,END=200) OLDFILE
      IF ( ( OLDFILE .NE. 1 ) .AND. ( OLDFILE .NE. 2 ) ) THEN
        GOTO 200
      ELSE IF ( OLDFILE .EQ. 1 ) THEN
*
*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME
        IERR = 0
        CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)
        IF ( IERR .EQ. 2 ) THEN
          PRINT*
          PRINT *, 'FILE ',PFN, ' NOT FOUND'
          PRINT*, 'PROGRAM ABORTED!!!'
          PRINT*
          PRINT*, 'FIND CORRECT BUILDING IDENTIFIER AND RESTART ',
     +              'PROGRAM'
          PRINT*
          STOP
*
        ELSE
          CALL LTYPE
          IF (TERR .NE. 0) CALL ERROR(5)
        END IF
      ELSE IF ( OLDFILE .EQ. 2 ) THEN
*
*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME
        IERR = 0
        CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)
        IF ( IERR .EQ. 0 ) THEN
          PRINT*
          PRINT*, 'DATA FILE ALREADY EXISTS FOR BUILDING ',BLDG
          PRINT*
          PRINT*,'IF YOU ENTER DATA AND STORE IT, YOU WILL WRITE ',
     +             'OVER THE OLD FILE.'
250       PRINT*
          PRINT*,'YOU MAY EITHER (1) ABORT OR (2) CONTINUE.'
          PRINT*,'INDICATE YOUR CHOICE BY ENTERING A NUMBER.'
          REWIND 1
          READ(1,*,END=250) ANSWER
            PRINT*
            PRINT*,'PROGRAM HAS BEEN ABORTED, PER YOUR REQUEST'
            PRINT*
```

```
              IF ( ANSWER .EQ. 1 ) THEN
                STOP
              ELSE IF ( ANSWER .EQ. 2 ) THEN
 255            CONTINUE
              ELSE
                GOTO 250
              END IF
            ELSE IF ( IERR .EQ. 2 ) THEN
*
*             NO DATA FILE EXISTS FOR THIS BUILDING AND DATA ENTRY
*             CAN CONTINUE
*
 260          CONTINUE
            ELSE
*
*             **PERMANENT FILE ERROR
*
              PRINT*
              PRINT*,'PROGRAM ABORTED !!!'
              PRINT*,'  SOME PERMANENT FILE ERROR HAS OCCURRED'
              PRINT*,'  DOUBLE CHECK YOUR BUILDING IDENTIFICATION ',
     +                 'AND TRY AGAIN'
              STOP
            END IF
            PRINT*
            PRINT*, ' BEGIN ENTERING DATA'
 300        TTOT = TTOT + 1
            IF ( TTOT .EQ. 1) THEN
              CALL DATAIN(1,TTOT)
            ELSE
              CALL DATAIN (0,TTOT)
            END IF
 400        PRINT*
            PRINT*, 'DO YOU WANT TO ENTER MORE DATA?',
     +                 '(1) YES  (2) NO'
            PRINT*, '  ENTER A NUMBEER !!!'
            REWIND 1
            READ(1,*,END=400) ANSWER
            IF ( (ANSWER .NE. 1) .AND. (ANSWER .NE. 2) ) THEN
              GOTO 400
            ELSE IF ( ANSWER .EQ. 1) THEN
              GOTO 300
            ELSE IF ( ANSWER .EQ. 2 ) THEN
              PRINT*
              PRINT*, 'DATA ENTRY DISCONTINUED'
            END IF
          END IF
*
*** MANIPULATE DATA
          CALL MANIP (QUIT,ABORT)
*
*** TERMINATE PROGRAM, STORING DATA IF NECESSARY
          IF ( QUIT .EQ. 1 ) THEN
           OPEN(UNIT=6,FILE=PFN(1:GETLEN(PFN)),FORM='FORMATTED',
     +            ACCESS='SEQUENTIAL',STATUS='NEW')
 500        FORMAT (1X,3(1X,A3),4(1X,F8.2))
            DO 600 N = 1,TTOT
              WRITE (6,500)(TYPE(N,Y1), Y1=1,3),(TDIM(N,Y2), Y2=1,4)
 600        CONTINUE
            ENDFILE(6)
            CALL PF ('REPLACE',0,PFN(1:GETLEN(PFN)))
WARNING* NUMBER OF ARGUMENTS IN REFERENCE TO _PF IS NOT CONSISTENT
            CLOSE(6,STATUS='DELETE')
            PRINT*
```

```
          PRINT*,'DATA HAS BEEN STORED AND PROGRAM TERMINATED'
        END IF
        IF( ABORT .EQ. 1 ) THEN
          PRINT*
          PRINT*, 'PROGRAM HAS BEEN ABORTED'
          PRINT*,'     NO DATA HAS BEEN STORED !!!'
        END IF
        STOP
        END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ABORT    1077B                            INTEGER
  AFLAG       2B  /INITILN/                 REAL
  ANSWER   1100B                            INTEGER
  BLDG        0B  /INITILC/                 CHAR*5
  FERR       66B  /INITILN/                 INTEGER
  FREQ        0B  /INITILN/                 REAL
  FREQA       4B  /INITILN/                 REAL           50
  FTOT       67B  /INITILN/                 INTEGER
  IERR     1105B                            INTEGER
  LINE     NONE            UNUSED/*S*       INTEGER
  N        1102B                            INTEGER
  NROOMS   1244B  /ROOMN/                   INTEGER
  OLDFILE  1101B                            INTEGER
  PFN      1106B                            CHAR*7
  QUALITY     1B  /INITILN/                 INTEGER
  QUIT     1076B                            INTEGER
  RAREA    1245B  /ROOMN/                   REAL           20
  RFLAG       3B  /INITILN/                 REAL
  ROOM        0B  /ROOMN/                   REAL          676
  TDBTOT    323B  /TYPEN/                   INTEGER
  TDB1       37B  /TYPEC/                   CHAR*3         35
  TDB2      215B  /TYPEN/                   REAL           70
  TDIM        0B  /TYPEN/                   REAL          140
  TERR      324B  /TYPEN/                   INTEGER
  TTOT      214B  /TYPEN/                   INTEGER
  TYPE        0B  /TYPEC/                   CHAR*3        105
  Y1       1103B                            INTEGER
  Y2       1104B                            INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-------------------------VALUE

  FMAX    INTEGER                         50
  RMAX    INTEGER                         20
  TMAX    INTEGER                         35


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  DATAIN                  2      SUBROUTINE        LTYPE                   0      SUBROUTINE
  ERROR                   1      SUBROUTINE        MANIP                   2      SUBROUTINE
  GETLEN     INTEGER      1      FUNCTION          PF                      5      SUBROUTINE
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF        -LABEL-ADDRESS-----PROPERTIES----DEF

   100      21B                     70          300     246B                    155
   200      47B                     82          400     260B                    161
   250     166B                    121          500     626B     FORMAT         184
   255 *NO REFS*                   132          600  INACTIVE    DO-TERM        187
   260 *NO REFS*                   141


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  STYPES     14B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ
  TAPE6   AUX/FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH           1112B =  586
  CM LABELLED COMMON LENGTH     1761B =  1009
  CM STORAGE USED              63000B =  26112
  COMPILE TIME                 0.287 SECONDS

       1  WARNING   ERROR IN  STYPES
```

```
      SUBROUTINE DATAIN (INSERT,LINE)                                    STYPES   151
*************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS           ***COMR       2
*************************************************************************COMR       3
      INTEGER RMAX                                                       COMR       4
      PARAMETER (RMAX = 20)                                              COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)       COMR       6
      INTEGER NROOMS                                                     COMR       7
      REAL ROOM                                                          COMR       8
*************************************************************************COMR       9
*************************************************************************COMR      10
*************************************************************************COMT       1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS               ***COMT       2
*************************************************************************COMT       3
      INTEGER TMAX                                                       COMT       4
      PARAMETER (TMAX=35)                                                COMT       5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR           COMT       6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                              COMT       7
      INTEGER TTOT,TDBTOT,TERR                                           COMT       8
      REAL TDIM,TDB2                                                     COMT       9
      CHARACTER * 3 TYPE,TDB1                                            COMT      10
*=================================================                       COMT      11
* DESCRIPTION OF ARRAYS                                                  COMT      12
*=================================================                       COMT      13
*   ID         MATERIAL    FRAME MATERIAL                                COMT      14
*-------------------------------------------------                       COMT      15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                       COMT      16
* A3          A3          A3                                             COMT      17
*=================================================                       COMT      18
* HEIGHT      WIDTH       LAYER       DISTANCE                           COMT      19
*                         THICKNESS   ABOVE FLOOR                        COMT      20
*-------------------------------------------------                       COMT      21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                          COMT      22
*  F8.2        F8.2        F8.2        F8.2                              COMT      23
*=================================================                       COMT      24
*  ID       ATTENUATION    AREA                                          COMT      25
* ------------------------------------------------                       COMT      26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                      COMT      27
*   A3          E9.3          E9.3                                       COMT      28
*************************************************************************COMT      29
*************************************************************************COMT      30
        INTEGER  ANSWER,LOK,DOK,NOK,GETLEN,VAL, INSERT,LINE,V            STYPES   154
        REAL H,W,T,ABOVE                                                 STYPES   155
        CHARACTER *3 DIR,ID,MAT,FMAT                                     STYPES   156
   99   IF ( INSERT .EQ. 1 ) THEN                                        STYPES   157
          ANSWER = 1                                                     STYPES   158
          INSERT = 1                                                     STYPES   159
        ELSE                                                             STYPES   160
100       PRINT*                                                         STYPES   161
          PRINT*, 'IS THIS THE FIRST LAYER OF A DOOR OR WINDOW?',        STYPES   162
     +              '  (1) YES (2) NO'                                   STYPES   163
          PRINT*, '      ENTER "0" TO ESCAPE "DATA ENTRY" MODE'          STYPES   164
          PRINT*, ' ENTER A NUMBER!!'                                    STYPES   165
          REWIND 1                                                       STYPES   166
          READ(1,*,END=100) ANSWER                                       STYPES   167
        END IF                                                           STYPES   168
*                                                                        STYPES   169
        IF (ANSWER .EQ. 0)  THEN                                         STYPES   170
          TTOT = TTOT - 1                                                STYPES   171
          PRINT*                                                         STYPES   172
          PRINT*, 'DATA ENTRY MODE ABORTED'                              STYPES   173
        END IF                                                           STYPES   174
*                                                                        STYPES   175
        IF ((ANSWER .NE. 2)                                              STYPES   176
```

```
     +     .AND. (ANSWER .NE. 1)
     +     .AND. (ANSWER .NE. 0)) THEN
          PRINT*
          PRINT*, 'INCORRECT NUMBER!!'
          PRINT*, ' TRY AGAIN!! -OR- ENTER "0" TO ESCAPE DATA ENTRY MODE'
          GOTO 99
        END IF
*
*     .................................
      IF ( ANSWER .EQ. 1 ) THEN
 300    PRINT*
        PRINT*, 'ENTER ''ID'' (E.G. ''WA'' OR ''DE'')'
        REWIND 1
        READ(1,*,END=300) ID
        IF (((ICHAR(ID(1:1)) .EQ. 55)
     +     .OR. (ICHAR(ID(1:1)) .EQ. 36))
     +     .AND. (ICHAR(ID(2:2)) .GE. 33)
     +     .AND. (ICHAR(ID(2:2)) .LE. 58)
     +     .AND. (GETLEN(ID) .EQ. 2)) THEN
          TYPE(LINE,1) = ID
        ELSE
          PRINT*
          PRINT*, 'INCORRECT ENTRY.  TRY AGAIN!!'
          GOTO 300
        END IF
*       ............................
 440      PRINT*
          PRINT*, 'ENTER HEIGHT, METERS'
          REWIND 1
          READ(1,*,END=440) TDIM(LINE,1)
*       ...............................
 460      PRINT*
          PRINT*, 'ENTER WIDTH, METERS'
          REWIND 1
          READ(1,*,END=460) TDIM(LINE,2)
*       ...............................
 470      PRINT*
          PRINT*, 'ENTER DISTANCE ABOVE FLOOR, METERS'
          REWIND 1
          READ(1,*,END=470) TDIM(LINE,4)
*       ...............................
 480      PRINT*
          PRINT*, 'ENTER THICKNESS OF LAYER, CENTIMETERS'
          REWIND 1
          READ(1,*,END=480) TDIM(LINE,3)
*       ...............................
 500      PRINT*
          PRINT*, 'ENTER "MATERIAL ID OF LAYER" (E.G. ''M01'')'
          REWIND 1
          READ(1,*,END=500) MAT
          IF ((GETLEN(MAT).EQ. 3)
     +       .AND. (MAT(1:1) .EQ. 'M')
     +       .AND. (ICHAR(MAT(2:2)) .GE. 16)
     +       .AND. (ICHAR(MAT(2:2)) .LE. 25)
     +       .AND. (ICHAR(MAT(3:3)) .GE. 16)
     +       .AND. (ICHAR(MAT(3:3)) .LE. 25)) THEN
            TYPE(LINE,2) = MAT
          ELSE
            PRINT*
            PRINT*, 'INCORRECT ENTRY!!  TRY AGAIN'
            GOTO 500
          END IF
*       ...............................
 510      PRINT*
```

```
          PRINT*, 'ENTER "MATERIAL ID OF FRAME" (E.G. ''M01'')'         STYPES    241
          REWIND 1                                                      STYPES    242
          READ(1,*,END=510) FMAT                                        STYPES    243
          IF ((GETLEN(FMAT).EQ. 3)                                      STYPES    244
     +       .AND. (FMAT(1:1) .EQ. 'M')                                 STYPES    245
     +       .AND. (ICHAR(FMAT(2:2)) .GE. 16)                           STYPES    246
     +       .AND. (ICHAR(FMAT(2:2)) .LE. 25)                           STYPES    247
     +       .AND. (ICHAR(FMAT(3:3)) .GE. 16)                           STYPES    248
     +       .AND. (ICHAR(FMAT(3:3)) .LE. 25)) THEN                     STYPES    249
            TYPE(LINE,3) = FMAT                                         STYPES    250
          ELSE                                                          STYPES    251
            PRINT*                                                      STYPES    252
            PRINT*, 'INCORRECT ENTRY!!  TRY AGAIN'                      STYPES    253
            GOTO 510                                                    STYPES    254
          END IF                                                        STYPES    255
        END IF                                                          STYPES    256
*                                                                       STYPES    257
*       ...................................                             STYPES    258
*                                                                       STYPES    259
        IF (ANSWER .EQ.2) THEN                                          STYPES    260
 580      PRINT*                                                        STYPES    261
          PRINT*, 'ENTER THICKNESS OF LAYER, CENTIMETERS'               STYPES    262
          REWIND 1                                                      STYPES    263
          READ(1,*,END=580) TDIM(LINE,3)                                STYPES    264
*          ..............................                               STYPES    265
 600      PRINT*                                                        STYPES    266
          PRINT*, 'ENTER "MATERIAL ID OF LAYER" (E.G. ''M01'')'         STYPES    267
          REWIND 1                                                      STYPES    268
          READ(1,*,END=600) MAT                                         STYPES    269
          IF ((GETLEN(MAT) .EQ. 3)                                      STYPES    270
     +       .AND. (MAT(1:1) .EQ. 'M')                                  STYPES    271
     +       .AND. (ICHAR(MAT(2:2)) .GE. 16)                            STYPES    272
     +       .AND. (ICHAR(MAT(2:2)) .LE. 25)                            STYPES    273
     +       .AND. (ICHAR(MAT(3:3)) .GE. 16)                            STYPES    274
     +       .AND. (ICHAR(MAT(3:3)) .LE. 25)) THEN                      STYPES    275
            TYPE(LINE,2) = MAT                                          STYPES    276
          ELSE                                                          STYPES    277
            PRINT*                                                      STYPES    278
            PRINT*, 'INCORRECT ENTRY!!    TRY AGAIN'                    STYPES    279
            GOTO 600                                                    STYPES    280
          END IF                                                        STYPES    281
          TYPE(LINE,3) = TYPE(LINE-1,3)                                 STYPES    282
          TYPE(LINE,1) = TYPE(LINE-1,1)                                 STYPES    283
          TDIM(LINE,1) = TDIM(LINE-1,1)                                 STYPES    284
          TDIM(LINE,2) = TDIM(LINE-1,2)                                 STYPES    285
          TDIM(LINE,4) = TDIM(LINE-1,4)                                 STYPES    286
        END IF                                                          STYPES    287
        RETURN                                                          STYPES    288
        END                                                             STYPES    289
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
ABOVE	NONE		UNUSED/*S*	REAL	
ANSWER	1076B			INTEGER	
DIR	NONE		UNUSED/*S*	CHAR*3	
DOK	NONE		UNUSED/*S*	INTEGER	
FMAT	1101B			CHAR*3	
H	NONE		UNUSED/*S*	REAL	
ID	1077B			CHAR*3	
INSERT	1	DUMMY-ARG		INTEGER	
LINE	2	DUMMY-ARG		INTEGER	
LOK	NONE		UNUSED/*S*	INTEGER	

```
 MAT       1100B                                  CHAR*3
 NOK       NONE                  UNUSED/*S*       INTEGER
 NROOMS    1244B   /ROOMN/                        INTEGER
 RAREA     1245B   /ROOMN/                        REAL           20
 ROOM         0B   /ROOMN/                        REAL          676
 T         NONE                  UNUSED/*S*       REAL
 TDBTOT     323B   /TYPEN/                        INTEGER
 TDB1        37B   /TYPEC/                        CHAR*3         35
 TDB2       215B   /TYPEN/                        REAL           70
 TDIM         0B   /TYPEN/                        REAL          140
 TERR       324B   /TYPEN/                        INTEGER
 TTOT       214B   /TYPEN/                        INTEGER
 TYPE         0B   /TYPEC/                        CHAR*3        105
 V         NONE                  UNUSED/*S*       INTEGER
 VAL       NONE                  UNUSED/*S*       INTEGER
 W         NONE                  UNUSED/*S*       REAL


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 RMAX    INTEGER                         20
 TMAX    INTEGER                         35


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 GETLEN    INTEGER         1      FUNCTION
 ICHAR     INTEGER         1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    99      7B                     45          480     213B                    106
   100     16B                     49          500     227B                    111
   300     70B                     75          510     311B                    128
   440    147B                     91          580     377B                    149
   460    163B                     96          600     413B                    154
   470    177B                    101


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 DATAIN      5B     2


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

 TAPE1   FMT/SEQ


--STATISTICS--

 PROGRAM-UNIT LENGTH           1104B =  580
 CM LABELLED COMMON LENGTH     1670B =  952
 CM STORAGE USED              63000B =  26112
 COMPILE TIME                 0.328 SECONDS
```

```
      SUBROUTINE MANIP (QUIT,ABORT)                                     STYPES     290
************************************************************************COMT         1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT         2
************************************************************************COMT         3
      INTEGER TMAX                                                      COMT         4
      PARAMETER (TMAX=35)                                               COMT         5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT         6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT         7
      INTEGER TTOT,TDBTOT,TERR                                          COMT         8
      REAL TDIM,TDB2                                                    COMT         9
      CHARACTER * 3 TYPE,TDB1                                           COMT        10
*==================================================                     COMT        11
* DESCRIPTION OF ARRAYS                                                 COMT        12
*==================================================                     COMT        13
*   ID        MATERIAL    FRAME MATERIAL                                COMT        14
*--------------------------------------------------                     COMT        15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                     COMT        16
* A3          A3          A3                                            COMT        17
*==================================================                     COMT        18
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT        19
*                         THICKNESS   ABOVE FLOOR                       COMT        20
*--------------------------------------------------                     COMT        21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT        22
*  F8.2        F8.2        F8.2        F8.2                             COMT        23
*==================================================                     COMT        24
*  ID       ATTENUATION    AREA                                         COMT        25
* -------------------------------------------------                     COMT        26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                     COMT        27
*   A3          E9.3         E9.3                                       COMT        28
************************************************************************COMT        29
************************************************************************COMT        30
      INTEGER ABORT,ANSWER,DOK,FLAG1,LOK,N,NOK,OK,OK1,OK2,QUIT,INSERT   STYPES     292
      INTEGER TEMP,V,X,Y,COMMAND                                        STYPES     293
*                                                                       STYPES     294
10    FLAG1 = 0                                                         STYPES     295
      PRINT*                                                            STYPES     296
      PRINT*, 'CHOOSE'                                                  STYPES     297
      PRINT*, '   (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES'      STYPES     298
      PRINT*, '   (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA'   STYPES     299
      PRINT*, '   (3) DELETE LINE           (6) STORE DATA AND EXIT ',  STYPES     300
     +         'PROGRAM'                                                STYPES     301
      PRINT*, '                             (7) EXIT PROGRAM WITHOUT ', STYPES     302
     +         'STORING DATA'                                           STYPES     303
      PRINT*, 'ENTER A NUMBER !!!'                                      STYPES     304
      PRINT*                                                            STYPES     305
      REWIND 1                                                          STYPES     306
      READ(1,*,END=10) COMMAND                                          STYPES     307
*                                                                       STYPES     308
*---------------------------------------------------------------------  STYPES     309
*** DISPLAY LINE ***                                                    STYPES     310
*---------------------------------------------------------------------  STYPES     311
      IF ( COMMAND .EQ. 1 ) THEN                                        STYPES     312
*                                                                       STYPES     313
*** INDICATE EMPTY DATA FILE                                            STYPES     314
        IF ( TTOT .EQ. 0 ) THEN                                         STYPES     315
          PRINT*                                                        STYPES     316
          PRINT*, 'DATA FILE IS EMPTY !!!'                              STYPES     317
*                                                                       STYPES     318
*** ENTER NUMBER OF LINE TO BE DISPLAYED                                STYPES     319
        ELSE                                                            STYPES     320
100       PRINT*                                                        STYPES     321
          PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED'      STYPES     322
          PRINT*, '     ( ENTER "0" TO ESCAPE DISPLAY MODE )'           STYPES     323
          REWIND 1                                                      STYPES     324
```

```
            READ(1,*,END=100) N
*
*** CHECK VALIDITY OF LINE NUMBER
            IF ( (N .GT. TTOT) .OR. (N .LT. 0) ) THEN
              PRINT*
              PRINT*, 'INCORRECT NUMBER !!!!!!       TRY AGAIN !!!'
              PRINT*, '             -OR-  ENTER "0" TO ESCAPE ID ',
     +                '"DISPLAY" MODE'
              GOTO 100
*
*** ABORT 'DISPLAY' MODE
            ELSE IF ( N .EQ. 0 ) THEN
              PRINT*
              PRINT*, '    "DISPLAY" MODE ABORTED !!!'
*
*** DISPLAY LINE OF DATA
            ELSE IF ( (N .GT. 0) .AND. (N .LE. TTOT) ) THEN
              PRINT*
              CALL DISPLAY( N, COMMAND)
*
            END IF
          END IF
        END IF
*
*----------------------------------------------------------------------
*** INSERT LINE ***
*----------------------------------------------------------------------
        IF ( COMMAND .EQ. 2 ) THEN
*
*** INDICATE EMPTY DATA FILE
          IF ( TTOT .EQ. 0 ) THEN
            PRINT*
            PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** REQUEST NUMBER OF LINE BEFORE WHICH INSERTION IS TO BE MADE
          ELSE
 200        PRINT*
            PRINT*, 'SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS ',
     +              'TO BE INSERTED'
            PRINT*, '     ( ENTER "0" TO ESCAPE "INSERTION" MODE )'
            REWIND 1
            READ(1,*,END=200) N
*
*** CHECK FOR VALID LINE NUMBER
            IF ( ( N .LT. 0 ) .OR. ( N .GT. TTOT ) ) THEN
              PRINT*
              PRINT*, 'INCORRECT NUMBER !!!'
              PRINT*, '      TRY AGAIN !!!  -OR-  ENTER "0" TO ESCAPE',
     +                '"INSERTION" MODE'
              GOTO 200
*
*** ABORT INSERTION MODE
            ELSE IF ( N .EQ. 0 ) THEN
              PRINT*
              PRINT*, '       "INSERTION" MODE ABORTED'
*
*** MAKE ROOM FOR NEW LINE OF DATA
            ELSE IF ( (N .GT. 0) .AND. (N .LE. TTOT) ) THEN
              DO 230 X = TTOT,N,-1
                DO 210 Y = 1,3
                  TYPE(X+1,Y) = TYPE(X,Y)
 210            CONTINUE
                DO 220 Y = 1,4
                  TDIM(X+1,Y) = TDIM(X,Y)
```

```
 220          CONTINUE                                             STYPES
 230        CONTINUE                                               STYPES
*                                                                  STYPES
*** ENTER DATA FOR NEW LINE                                        STYPES
            TTOT = TTOT + 1                                        STYPES
            CALL DATAIN (1,N)                                      STYPES
*                                                                  STYPES
*** INITIALIZE FLAGS                                               STYPES
            OK1 = 0                                                STYPES
            OK2 = 0                                                STYPES
            OK = 0                                                 STYPES
*                                                                  STYPES
*** TEST VALIDITY OF DATA                                          STYPES
*                                                                  STYPES
***TEST IF NEW LAYER BELONGS TO THE NEXT TYPE                      STYPES
            IF ( ( TYPE(N,1) .EQ. TYPE(N+1,1) )                    STYPES
     +           .AND. ( TYPE(N,3) .EQ. TYPE(N+1,3) ) ) THEN       STYPES
              IF ( ( TDIM(N,1) .EQ. TDIM(N+1,1) )                  STYPES
     +             .AND. ( TDIM(N,2) .EQ. TDIM(N+1,2) )            STYPES
     +             .AND. ( TDIM(N,4) .EQ. TDIM(N+1,4) ) ) THEN     STYPES
                OK1 = 1                                            STYPES
              END IF                                               STYPES
            END IF                                                 STYPES
*                                                                  STYPES
*** TEST IF NEW LAYER BELONGS TO PREVIOUS TYPE                     STYPES
            IF ( N .GT. 1 ) THEN                                   STYPES
              IF ( ( TYPE(N,1) .EQ. TYPE(N-1,1) )                  STYPES
     +             .AND. ( TYPE(N,3) .EQ. TYPE(N-1,3) ) ) THEN     STYPES
                IF ( ( TDIM(N,1) .EQ. TDIM(N-1,1) )                STYPES
     +               .AND. ( TDIM(N,2) .EQ. TDIM(N-1,2) )          STYPES
     +               .AND. ( TDIM(N,4) .EQ. TDIM(N-1,4) ) ) THEN   STYPES
                  OK2 = 1                                          STYPES
                END IF                                             STYPES
              END IF                                               STYPES
            END IF                                                 STYPES
*                                                                  STYPES
            IF ( ( OK1 .EQ. 1 ) .OR. ( OK2 .EQ. 1 ) ) THEN         STYPES
              OK = 1                                               STYPES
            END IF                                                 STYPES
*                                                                  STYPES
            IF ( OK .EQ. 1 ) THEN                                  STYPES
              PRINT*                                               STYPES
              PRINT*, 'THE FOLLOWING LINE HAS BEEN ADDED AS LINE ', N  STYPES
              CALL DISPLAY( N, COMMAND)                            STYPES
*                                                                  STYPES
*** REJECT DATA IF DATA DOESN'T MATCH PREVIOUS OR NEXT LAYER       STYPES
            ELSE IF ( OK .EQ. 0 ) THEN                             STYPES
              PRINT*                                               STYPES
              PRINT*, 'YOUR DATA WAS NOT ACCEPTED !!!'             STYPES
              PRINT*, 'YOUR DATA MUST REPRESENT A LAYER ',         STYPES
     +                'IN AN EXISTING DOOR OR WINDOW'              STYPES
              PRINT*, ' I.E.  THE ID, FRAME MATERIAL, HEIGHT, ',   STYPES
     +               'WIDTH, AND DISTANCE ABOVE FLOOR'             STYPES
              PRINT*, '        PARAMETERS MUST MATCH THE DOOR OR ',  STYPES
     +                'WINDOW JUST BEFORE'                         STYPES
              PRINT*, '        OR JUST AFTER YOUR SPECIFIED INSERTION ',STYPES
     +                'POINT'                                      STYPES
              PRINT*                                               STYPES
              PRINT*, 'THE FOLLOWING DISPLAYS'                     STYPES
              IF ( N .GT. 1 ) PRINT*,'THE LINE BEFORE YOUR LINE,'  STYPES
              PRINT*, 'YOUR LINE, AND THE LINE AFTER'              STYPES
              PRINT*                                               STYPES
*                                                                  STYPES
*** DISPLAY LINES OF DATA                                          STYPES
```

```
                  IF ( N .GT. 1 ) CALL DISPLAY ( N-1, COMMAND )
                  CALL DISPLAY( N, COMMAND)
                  CALL DISPLAY ( N+1, COMMAND)
*
*** REMOVE THE LINE OF INCORRECTLY ENTERED DATA
                  DO 270  X = N,TTOT
                    DO 250  Y = 1,3
                      TYPE(X,Y) = TYPE(X+1,Y)
250                 CONTINUE
                    DO 260  Y = 1,4
                      TDIM(X,Y) = TDIM(X+1,Y)
260                 CONTINUE
270               CONTINUE
                  TTOT = TTOT - 1
                END IF
              END IF
            END IF
          END IF
*
*--------------------------------------------------------------------
*** DELETE LINE ***
*--------------------------------------------------------------------
          IF ( COMMAND .EQ. 3 ) THEN
*
*** INDICATE EMPTY DATA FILE
            IF ( TTOT .EQ. 0 ) THEN
              PRINT*
              PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** READ NUMBER OF LINE TO BE DELETED
            ELSE
300           PRINT*
              PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DELETED'
              PRINT*, '   (ENTER "0" TO ESCAPE DELETION MODE)'
              REWIND 1
              READ(1,*,END=300) N
*
*** CHECK VALIDITY OF LINE NUMBER
              IF ( (N .GT. TTOT ) .OR. ( N .LT. 0 ) ) THEN
                PRINT*
                PRINT*, ' INCORRECT NUMBER !!!'
                PRINT*, '    TRY AGAIN !!!  -OR-  ENTER "0" TO ESCAPE ID',
     +               '"DELETE" MODE'
                GOTO 300
*
*** ABORT 'DELETE' MODE
              ELSE IF ( N .EQ. 0 ) THEN
                PRINT*, '  "DELETE" MODE ABORTED'
*
*** DOUBLE CHECK CHOICE OF LINE TO BE DELETED
              ELSE IF (( N .GT. 0 ) .AND. ( N .LE. TTOT )) THEN
                PRINT*
                PRINT*, 'DOUBLE CHECK !!!'
                PRINT*, '     DO YOU WANT TO DELETE THE FOLLOWING LINE?:'
                CALL DISPLAY( N, COMMAND)
305             PRINT*, '   ENTER (1) YES  OR   (2) NO'
                REWIND 1
                READ(1,*,END=305) ANSWER
*
*** DELETE LINE
                IF ( ANSWER .EQ. 1 ) THEN
                  DO 330 X = N, TTOT - 1
                    DO 310 Y = 1,3
                      TYPE(X,Y) = TYPE(X+1,Y)
```

```
310             CONTINUE
                DO 320 Y = 1,4
                  TDIM(X,Y) = TDIM(X+1,Y)
320             CONTINUE
330           CONTINUE
              TTOT = TTOT - 1
              PRINT*
              PRINT*, 'LINE # ',N,' DELETED'
            END IF
*
          END IF
        END IF
      END IF
*
*-----------------------------------------------------------------------------
*** DISPLAY ALL DATA ***
*-----------------------------------------------------------------------------
      IF ( COMMAND .EQ. 4 ) THEN
*
*** INDICATE EMPTY DATA FILE
        IF ( TTOT .EQ. 0 ) THEN
          PRINT*
          PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** DISPLAY DATA
        ELSE
          PRINT*
          CALL DISPLAY( N, COMMAND)
*
        END IF
      END IF
*
*-----------------------------------------------------------------------------
*** ADD DATA ***
*-----------------------------------------------------------------------------
      IF ( COMMAND .EQ. 5 ) THEN
*
*** ENTER DATA
500     TTOT = TTOT + 1
*
        CALL DATAIN (0,TTOT)
510     PRINT*
        PRINT*, 'DO YOU WANT TO ENTER MORE DATA?   (1) YES  (2) NO'
        PRINT*, '   ENTER A NUMBER !!!'
        REWIND 1
        READ(1,*,END=510) ANSWER
*
*** CHECK VALIDITY OF NUMBER
*
*** ENTER MORE DATA
        IF ( ANSWER .EQ. 1 ) THEN
          GOTO 500
*
*** DISCONTINUE DATA ENTRY
        ELSE IF ( ANSWER .EQ. 2 ) THEN
          PRINT*
          PRINT*, 'DATA ENTRY DISCONTINUED'
*
*** INVALID ENTRY
        ELSE
          GOTO 510
        END IF
      END IF
*
```

```
*----------------------------------------------------------------------
*** STORE DATA AND PROGRAM ***
*----------------------------------------------------------------------
      IF ( COMMAND .EQ. 6 ) THEN
 600    PRINT*
        PRINT*, 'DOUBLE CHECK !!!'
        PRINT*, '    DO YOU YOU WANT TO STORE THIS DATA AND END PROG'
        PRINT*, ' NOTE. STORING THIS DATA WILL WIPE OUT ANY OLD FILE '
        PRINT*, ' OF THE SAME NAME !!!'
        PRINT*, '         ENTER A NUMBER: (1) YES   (2) NO'
        REWIND 1
        READ(1,*,END=600) ANSWER
*
*** SET FLAG FOR STORING DATA IN THE MAIN PROGRAM
        IF ( ANSWER .EQ. 1 ) THEN
          QUIT = 1
          RETURN
*
*** ABORT 'STORING' MODE
        ELSE IF ( ANSWER .EQ. 2 ) THEN
          PRINT*
          PRINT*, '    "STORING" MODE DISCONTINUED'
*
*** CHECK VALIDITY OF ANSWER
        ELSE IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN
          GOTO 600
*
        END IF
      END IF
*
*----------------------------------------------------------------------
*** END PROGRAM WITHOUT STORING DATA ***
*----------------------------------------------------------------------
      IF ( COMMAND .EQ. 7 ) THEN
 700    PRINT*
        PRINT*, 'DOUBLE CHECK !!!'
        PRINT*, '    DO YOU WANT TO END THIS PROGRAM ',
     +         'WITHOUT STORING DATA?'
        PRINT*, '         ENTER A NUMBER: (1) YES   (2) NO'
        REWIND 1
        READ(1,*,END=700) ANSWER
*
*** SET FLAG FOR ABORTING PROGRAM IN THE MAIN PROGRAM
        IF ( ANSWER .EQ. 1 ) THEN
          ABORT = 1
          RETURN
*
*** ABORT 'STORING' MODE
        ELSE IF ( ANSWER .EQ. 2 ) THEN
          PRINT*
          PRINT*, '    "ABORTION" MODE DISCONTINUED'
*
*** CHECK VALIDITY OF ANSWER
        ELSE IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN
          GOTO 700
*
        END IF
      END IF
*
*----------------------------------------------------------------------
*** LOOP TO BEGINNING OF 'MANIP' SUBROUTINE
*----------------------------------------------------------------------
      GOTO 10
*
```

```
    385       RETURN                                              STYPES     645
TRIVIAL* NO PATH TO THIS STATEMENT
    386       END                                                 STYPES     646
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
ABORT	2	DUMMY-ARG		INTEGER	
ANSWER	2030B			INTEGER	
COMMAND	2040B			INTEGER	
DOK	NONE		UNUSED/*S*	INTEGER	
FLAG1	2031B			INTEGER	
INSERT	NONE		UNUSED/*S*	INTEGER	
LOK	NONE		UNUSED/*S*	INTEGER	
N	2032B			INTEGER	
NOK	NONE		UNUSED/*S*	INTEGER	
OK	2033B			INTEGER	
OK1	2034B			INTEGER	
OK2	2035B			INTEGER	
QUIT	1	DUMMY-ARG		INTEGER	
TDBTOT	323B	/TYPEN/		INTEGER	
TDB1	37B	/TYPEC/		CHAR*3	35
TDB2	215B	/TYPEN/		REAL	70
TDIM	0B	/TYPEN/		REAL	140
TEMP	NONE		UNUSED/*S*	INTEGER	
TERR	324B	/TYPEN/		INTEGER	
TTOT	214B	/TYPEN/		INTEGER	
TYPE	0B	/TYPEC/		CHAR*3	105
V	NONE		UNUSED/*S*	INTEGER	
X	2036B			INTEGER	
Y	2037B			INTEGER	

--SYMBOLIC CONSTANTS--(LO=A)

-NAME----	TYPE------------------------	VALUE
TMAX	INTEGER	35

--PROCEDURES--(LO=A)

-NAME------	TYPE--------	ARGS------	CLASS-----
DATAIN		2	SUBROUTINE
DISPLAY		2	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL-	ADDRESS-----	PROPERTIES----	DEF	-LABEL-	ADDRESS-----	PROPERTIES----	DEF
10	7B		35	300	547B		224
100	50B		61	305	616B		248
200	133B		101	310	INACTIVE	DO-TERM	257
210	INACTIVE	DO-TERM	126	320	INACTIVE	DO-TERM	260
220	INACTIVE	DO-TERM	129	330	INACTIVE	DO-TERM	261
230	INACTIVE	DO-TERM	130	500	741B		295
250	INACTIVE	DO-TERM	201	510	745B		298
260	INACTIVE	DO-TERM	204	600	1001B		325
270	INACTIVE	DO-TERM	205	700	1053B		355

```
SUBROUTINE MANIP    74/175  OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  MANIP       5B     2


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH           2054B =  1068
  CM LABELLED COMMON LENGTH      377B =  255
  CM STORAGE USED             63000B =  26112
  COMPILE TIME                0.553 SECONDS

      1  TRIVIAL   ERROR IN  MANIP
```

```
      SUBROUTINE DISPLAY (LINE, COMMAND)                                STYPES     647
************************************************************************COMT         1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT         2
************************************************************************COMT         3
      INTEGER TMAX                                                      COMT         4
      PARAMETER (TMAX=35)                                               COMT         5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT         6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT         7
      INTEGER TTOT,TDBTOT,TERR                                          COMT         8
      REAL TDIM,TDB2                                                    COMT         9
      CHARACTER * 3 TYPE,TDB1                                           COMT        10
*==================================================                     COMT        11
* DESCRIPTION OF ARRAYS                                                 COMT        12
*==================================================                     COMT        13
*   ID        MATERIAL    FRAME MATERIAL                                COMT        14
*--------------------------------------------------                     COMT        15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                     COMT        16
* A3          A3          A3                                            COMT        17
*==================================================                     COMT        18
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT        19
*                         THICKNESS   ABOVE FLOOR                       COMT        20
*--------------------------------------------------                     COMT        21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT        22
*  F8.2        F8.2        F8.2        F8.2                             COMT        23
*==================================================                     COMT        24
*  ID      ATTENUATION    AREA                                          COMT        25
* -------------------------------------------------                     COMT        26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                     COMT        27
*   A3          E9.3         E9.3                                       COMT        28
************************************************************************COMT        29
************************************************************************COMT        30
      INTEGER LINE, COMMAND, N                                          STYPES     649
1000  FORMAT (1X,A5,2X,A2,2X,A8,2X,A8,2X,A11,2X,A9,2X,A8,2X,A8)         STYPES     650
2000  FORMAT (2X,I3,3X,A2,3X,F6.2,4X,F6.2,4X,F6.2,7X,F6.2,6X,A3,        STYPES     651
     +       7X,A3)                                                     STYPES     652
      PRINT 1000, 'LINE','ID','HEIGHT  ','WIDTH   ','DISTANCE   ',      STYPES     653
     +             'THICKNESS','LAYER   ','FRAME   '                    STYPES     654
      PRINT 1000,' ',' ','(METERS)','(METERS)','ABOVE FLOOR',           STYPES     655
     +            '(CM)      ','MATERIAL','MATERIAL'                    STYPES     656
      IF ( COMMAND .EQ. 4 ) THEN                                        STYPES     657
        DO 10 LINE = 1,TTOT                                             STYPES     658
          PRINT 2000, LINE,TYPE(LINE,1),TDIM(LINE,1),TDIM(LINE,2),      STYPES     659
     +         TDIM(LINE,4),TDIM(LINE,3),TYPE(LINE,2),TYPE(LINE,3)      STYPES     660
10      CONTINUE                                                        STYPES     661
      ELSE                                                              STYPES     662
          PRINT 2000, LINE,TYPE(LINE,1),TDIM(LINE,1),TDIM(LINE,2),      STYPES     663
     +         TDIM(LINE,4),TDIM(LINE,3),TYPE(LINE,2),TYPE(LINE,3)      STYPES     664
      END IF                                                            STYPES     665
      RETURN                                                            STYPES     666
      END                                                               STYPES     667
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  COMMAND      2   DUMMY-ARG                 INTEGER
  LINE         1   DUMMY-ARG                 INTEGER
  N          NONE              UNUSED/*S*    INTEGER
  TDBTOT     323B  /TYPEN/                   INTEGER
  TDB1        37B  /TYPEC/                   CHAR*3          35
  TDB2       215B  /TYPEN/                   REAL            70
  TDIM        0B   /TYPEN/                   REAL           140
  TERR       324B  /TYPEN/                   INTEGER
  TTOT       214B  /TYPEN/                   INTEGER
```

```
 TYPE         0B   /TYPEC/                          CHAR*3          105


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 TMAX     INTEGER                          35


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        44
   1000     163B    FORMAT         33
   2000     171B    FORMAT         34


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DISPLAY     5B      2


--STATISTICS--

  PROGRAM-UNIT LENGTH           254B =  172
  CM LABELLED COMMON LENGTH     377B =  255
  CM STORAGE USED             61000B =  25088
  COMPILE TIME                0.100 SECONDS
```

```
      INTEGER FUNCTION VAL(STRING)                              STYPES     668
C** RETURNS THE INTEGER VALUE OF A STRING.                      STYPES     669
      INTEGER NUMBER, X,L,EXP,DIGIT,GETLEN                      STYPES     670
      CHARACTER * (*) STRING                                    STYPES     671
      L = GETLEN(STRING)                                        STYPES     672
      NUMBER = 0                                                STYPES     673
      DO 10 X = L,1,-1                                          STYPES     674
      EXP = L - X                                               STYPES     675
      DIGIT = ICHAR(STRING(X:X)) - 16                           STYPES     676
      NUMBER = NUMBER + DIGIT*10**EXP                           STYPES     677
 10   CONTINUE                                                  STYPES     678
      VAL = NUMBER                                              STYPES     679
      RETURN                                                    STYPES     680
      END                                                       STYPES     681
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  DIGIT      76B                                  INTEGER
  EXP        75B                                  INTEGER
  L          74B                                  INTEGER
  NUMBER     72B                                  INTEGER
  STRING      1    DUMMY-ARG                      CHAR*(*)
  VAL        71B                                  INTEGER
  X          73B                                  INTEGER


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN     INTEGER        1       FUNCTION
  ICHAR      INTEGER        1       INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        11


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  VAL          6B     1


--STATISTICS--

  PROGRAM-UNIT LENGTH           102B =   66
  CM STORAGE USED             61000B =   25088
  COMPILE TIME                0.041 SECONDS
```

```
FTN 5.1+552        83/12/24. 10.36.12 PAGE     21
  FUNCTION GETLEN   74/175  OPT=0

      INTEGER FUNCTION GETLEN (STRING)
C
C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING
C
C
C ARGUMENT DEFINITIONS --
C   READ ARGUMENTS
C     STRING - STRING WHOSE LENGTH IS TO BE DETERMINED
C
      CHARACTER * (*) STRING
C
C FUNCTION PARAMETERS
      CHARACTER * 1 BLANK
      PARAMETER (BLANK = ' ')
C
C LOCAL VARIABLES
      INTEGER NEXT
C
C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK
      DO 10 NEXT = LEN(STRING),1,-1
         IF (STRING(NEXT : NEXT) .NE. BLANK) THEN
            GETLEN = NEXT
            RETURN
         END IF
   10 CONTINUE
C
C ALL CHARACTERS ARE BLANKS
      GETLEN = 0
C
      RETURN
      END

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  GETLEN     63B                             INTEGER
  NEXT       64B                             INTEGER
  STRING       1   DUMMY-ARG                 CHAR*(*)

--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  BLANK   CHAR*1                      ' '

--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  LEN       INTEGER        1      INTRINSIC

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        25
```

```
FTN 5.1+552        83/12/24. 10.36.12 PAGE     22
  FUNCTION GETLEN   74/175  OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  GETLEN      6B      1

--STATISTICS--

  PROGRAM-UNIT LENGTH           70B =    56
  CM STORAGE USED            61000B = 25088
  COMPILE TIME               0.041 SECONDS
```

```
      SUBROUTINE LTYPE                                                  LTYPE      1
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LTYPE      2
*!!!                                                                 !!!LTYPE      3
*!!!   LOAD THE "TYPE" ARRAYS FROM THE TYPE DATA FILE                !!!LTYPE      4
*!!!                                                                 !!!LTYPE      5
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LTYPE      6
************************************************************************LTYPE      7
************************************************************************COMT       1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT       2
************************************************************************COMT       3
      INTEGER TMAX                                                      COMT       4
      PARAMETER (TMAX=35)                                               COMT       5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT       6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT       7
      INTEGER TTOT,TDBTOT,TERR                                          COMT       8
      REAL TDIM,TDB2                                                    COMT       9
      CHARACTER * 3 TYPE,TDB1                                           COMT      10
*===============================================                        COMT      11
* DESCRIPTION OF ARRAYS                                                 COMT      12
*===============================================                        COMT      13
*   ID        MATERIAL    FRAME MATERIAL                                COMT      14
*-----------------------------------------------                        COMT      15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                      COMT      16
* A3          A3          A3                                            COMT      17
*===============================================                        COMT      18
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT      19
*                         THICKNESS   ABOVE FLOOR                       COMT      20
*-----------------------------------------------                        COMT      21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT      22
*  F8.2         F8.2        F8.2        F8.2                            COMT      23
*===============================================                        COMT      24
*  ID       ATTENUATION    AREA                                         COMT      25
* ----------------------------------------------                        COMT      26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                     COMT      27
*   A3          E9.3         E9.3                                       COMT      28
************************************************************************COMT      29
************************************************************************COMT      30
************************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF       2
************************************************************************COMF       3
      INTEGER FMAX                                                      COMF       4
      PARAMETER (FMAX = 50)                                             COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF       6
     $                 FTOT                                             COMF       7
      COMMON /INITILC/ BLDG                                             COMF       8
      CHARACTER * 5 BLDG                                                COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF      10
      INTEGER QUALITY, FERR, FTOT                                       COMF      11
************************************************************************COMF      12
************************************************************************COMF      13
************************************************                        LTYPE     10
*   DECLARATION OF VARIABLES                                            LTYPE     11
************************************************                        LTYPE     12
      INTEGER GETLEN, R, C                                              LTYPE     13
      CHARACTER * 7  PFN                                                LTYPE     14
************************************************                        LTYPE     15
*                                                                       LTYPE     16
************************************************                        LTYPE     17
      PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'T'                          LTYPE     18
      TERR = 0                                                          LTYPE     19
      CALL  PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',TERR)                   LTYPE     20
      IF (TERR .EQ. 0 ) THEN                                            LTYPE     21
        OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',                       LTYPE     22
     $        STATUS='OLD', ACCESS='SEQUENTIAL')                        LTYPE     23
```

```
FTN 5.1+552          83/12/24. 10.36.12 PAGE     24
SUBROUTINE LTYPE      74/175   OPT=0

     65 1000     FORMAT (1X,3(1X,A3),4(1X,F8.2))                                    LTYPE        24
     66          TTOT = 0                                                           LTYPE        25
     67          DO 10 R = 1,TMAX                                                   LTYPE        26
     68            READ (3,1000,END=20)(TYPE(R,C),C=1,3),(TDIM(R,C),C=1,4)          LTYPE        27
     69            TTOT = TTOT + 1                                                  LTYPE        28
     70 10       CONTINUE                                                           LTYPE        29
     71 20       CONTINUE                                                           LTYPE        30
     72          CLOSE(3,STATUS='DELETE')                                           LTYPE        31
     73        ELSE IF ( TERR .EQ. 2 ) THEN                                         LTYPE        32
     74          CALL WARNING (5)                                                   LTYPE        33
     75        ELSE                                                                 LTYPE        34
     76          CALL WARNING (6)                                                   LTYPE        35
     77        END IF                                                               LTYPE        36
     78        RETURN                                                               LTYPE        37
     79        END                                                                  LTYPE        38
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG        2B  /INITILN/                  REAL
  BLDG         0B  /INITILC/                  CHAR*5
  C          236B                             INTEGER
  FERR        66B  /INITILN/                  INTEGER
  FREQ         0B  /INITILN/                  REAL
  FREQA        4B  /INITILN/                  REAL            50
  FTOT        67B  /INITILN/                  INTEGER
  PFN        237B                             CHAR*7
  QUALITY      1B  /INITILN/                  INTEGER
  R          235B                             INTEGER
  RFLAG        3B  /INITILN/                  REAL
  TDBTOT     323B  /TYPEN/                    INTEGER
  TDB1        37B  /TYPEC/                    CHAR*3          35
  TDB2       215B  /TYPEN/                    REAL            70
  TDIM         0B  /TYPEN/                    REAL           140
  TERR       324B  /TYPEN/                    INTEGER
  TTOT       214B  /TYPEN/                    INTEGER
  TYPE         0B  /TYPEC/                    CHAR*3         105
```

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  FMAX    INTEGER                        50
  TMAX    INTEGER                        35
```

```
--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN    INTEGER          1       FUNCTION
  PF                         5       SUBROUTINE
  WARNING                    1       SUBROUTINE
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE    DO-TERM         70
    20    111B                      71
  1000    147B      FORMAT          65
```

```
FTN 5.1+552          83/12/24. 10.36.12 PAGE     25
SUBROUTINE LTYPE      74/175   OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  LTYPE       5B      0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE3   AUX/FMT/SEQ

--STATISTICS--

  PROGRAM-UNIT LENGTH              245B =   165
  CM LABELLED COMMON LENGTH        470B =   312
  CM STORAGE USED                63000B = 26112
  COMPILE TIME                   0.107 SECONDS
```

```
FTN 5 1+552          83/12/24. 10.36.12 PAGE     26
SUBROUTINE ERROR      74/175   OPT=0

          SUBROUTINE ERROR(IERR)                                           ERROR        1
          CHARACTER*45 MESSAGE(20)                                         ERROR        2
          DATA MESSAGE( 1)/'MATERIALS DATA BASE IS EMPTY                 '/ ERROR        3
          DATA MESSAGE( 2)/'FREQUENCY IS OUT OF RANGE                    '/ ERROR        4
          DATA MESSAGE( 3)/'THIS MATERIAL IS NOT IN DATA BASE            '/ ERROR        5
          DATA MESSAGE( 4)/'DENOMINATOR IS ZERO                          '/ ERROR        6
          DATA MESSAGE( 5)/'FILE HANDLING ERROR                          '/ ERROR        7
          DATA MESSAGE( 6)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR        8
          DATA MESSAGE( 7)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR        9
          DATA MESSAGE( 8)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       10
          DATA MESSAGE( 9)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       11
          DATA MESSAGE(10)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       12
          DATA MESSAGE(11)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       13
          DATA MESSAGE(12)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       14
          DATA MESSAGE(13)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       15
          DATA MESSAGE(14)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       16
          DATA MESSAGE(15)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       17
          DATA MESSAGE(16)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       18
          DATA MESSAGE(17)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       19
          DATA MESSAGE(18)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       20
          DATA MESSAGE(19)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       21
          DATA MESSAGE(20)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       22
          IERRM=5                                                          ERROR       23
          IF(IERR.GT.IERRM) IERR=20                                        ERROR       24
          WRITE(6,10) IERR,MESSAGE(IERR)                                   ERROR       25
   10     FORMAT(' ***ERROR NUMBER = ',I5,' ***  ',A45)                    ERROR       26
          CALL PMDSTOP                                                     ERROR       27
          STOP 'ERROR'                                                     ERROR       28
          END                                                              ERROR       29
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  IERR        1   DUMMY-ARG                  INTEGER
  IERRM     210B                             INTEGER
  MESSAGE    56B                             CHAR*45        20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  PMDSTOP                  0      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10      36B     FORMAT          26


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ERROR       5B      1
```

```
FTN 5.1+552          83/12/24. 10.36.12 PAGE     27
SUBROUTINE ERROR      74/175   OPT=0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE6    FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH                213B =   139
  CM STORAGE USED                 61000B =  25088
  COMPILE TIME                     0.054 SECONDS
```

```
 1          SUBROUTINE WARNING(ERR)                                              WARNING       1
 2          INTEGER ERR, ERRM                                                    WARNING       2
 3          CHARACTER*45 MESSAGE(20)                                             WARNING       3
 4          DATA MESSAGE( 1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/  WARNING       4
 5          DATA MESSAGE( 2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE    '/  WARNING       5
 6          DATA MESSAGE( 3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG   '/  WARNING       6
 7          DATA MESSAGE( 4)/'FILE HANDLING PROBLEM ON "MATTER FILE        '/  WARNING       7
 8          DATA MESSAGE( 5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/  WARNING       8
 9          DATA MESSAGE( 6)/'FILE HANDLING PROBLEM ON "TYPE" FILE         '/  WARNING       9
10          DATA MESSAGE( 7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG'/  WARNING      10
11          DATA MESSAGE( 8)/'FILE HANDLING PROBLEM ON "WALL" FILE         '/  WARNING      11
12          DATA MESSAGE( 9)/'HEIGHT AND WIDTH OF ROOM MISSING             '/  WARNING      12
13          DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING                    '/  WARNING      13
14          DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG       '/  WARNING      14
15          DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE         '/  WARNING      15
16          DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE                 '/  WARNING      16
17          DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE                 '/  WARNING      17
18          DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE                 '/  WARNING      18
19          DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE                 '/  WARNING      19
20          DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE                 '/  WARNING      20
21          DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE                 '/  WARNING      21
22          DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE                 '/  WARNING      22
23          DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE                 '/  WARNING      23
24          ERRM=12                                                              WARNING      24
25          IERR = ERR                                                           WARNING      25
26          IF(ERR.GT.ERRM) IERR=20                                              WARNING      26
27          WRITE(6,20)                                                          WARNING      27
28          WRITE(6,10) ERR,MESSAGE(IERR)                                        WARNING      28
29          WRITE(6,20)                                                          WARNING      29
30 10       FORMAT(' ***WARNING NUMBER = ',I5,' ***  ',A45)                      WARNING      30
31 20       FORMAT('      ')                                                     WARNING      31
32          RETURN                                                               WARNING      32
33          END                                                                  WARNING      33
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ERR          1   DUMMY-ARG                  INTEGER
  ERRM        60B                             INTEGER
  IERR       213B                             INTEGER
  MESSAGE     61B                             CHAR*45         20


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10      34B     FORMAT          30
     20      42B     FORMAT          31


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  WARNING      5B      1
```



```
--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE6   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH             216B =   142
  CM STORAGE USED               61000B = 25088
  COMPILE TIME                  0.058 SECONDS
```

Appendix 9.6 Listing of Computer Program SFREQ

```
      PROGRAM SFREQ (INPUT,TAPE1=INPUT)                                SFREQ      1
*                                                                      SFREQ      2
*THIS INTERACTIVE PROGRAM INPUTS THE DATA DESCRIBING EACH FREQ         SFREQ      3
*IN THE BUILDING AND STORES IT.  THE FILE NAME IS CREATED BY           SFREQ      4
*ATTACHING "B" TO THE FRONT OF AND "F" TO THE BACK OF THE BUILDING     SFREQ      5
*IDENTIFICATION. THE BUILDING IDENTIFICATION CAN BE NO MORE            SFREQ      6
*THAN 5 ALPHANUMERIC CHARACTERS.                                       SFREQ      7
                                                                       SFREQ      8
***********************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                   ***COMF       2
***********************************************************************COMF       3
      INTEGER FMAX                                                     COMF       4
      PARAMETER (FMAX = 50)                                            COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF       6
     $                   FTOT                                          COMF       7
      COMMON /INITILC/ BLDG                                            COMF       8
      CHARACTER * 5 BLDG                                               COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                   COMF      10
      INTEGER QUALITY, FERR, FTOT                                      COMF      11
***********************************************************************COMF      12
***********************************************************************COMF      13
      INTEGER GETLEN,QUIT,ABORT,ANSWER,OLDFILE,N,Y1,Y2,LINE            SFREQ     10
      INTEGER IERR                                                     SFREQ     11
      CHARACTER * 7  PFN                                               SFREQ     12
*                                                                      SFREQ     13
* INITIALIZATION                                                       SFREQ     14
      QUIT = 0                                                         SFREQ     15
      FTOT = 0                                                         SFREQ     16
      ABORT = 0                                                        SFREQ     17
100   PRINT*                                                           SFREQ     18
      PRINT *, 'ENTER BUILDING IDENTIFICATION  (E.G. ''101'')'         SFREQ     19
      PRINT *, '    (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)'          SFREQ     20
      REWIND 1                                                         SFREQ     21
      READ(1,*,END=100) BLDG                                           SFREQ     22
                                                                       SFREQ     23
      IF ( GETLEN(BLDG) .GT. 5 ) THEN                                  SFREQ     24
        GO TO 100                                                      SFREQ     25
      END IF                                                           SFREQ     26
      PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'F'                         SFREQ     27
*                                                                      SFREQ     28
*** LOAD DATA FROM EXISTING FILE IF NECESSARY                          SFREQ     29
200   PRINT*                                                           SFREQ     30
      PRINT*,'WILL THIS BE'                                            SFREQ     31
      PRINT*,'   (1) A MODIFICATION OF AN EXISTING FILE?'              SFREQ     32
      PRINT*,'   (2) A NEW FILE?'                                      SFREQ     33
      PRINT*,'ENTER A NUMBER  !!!'                                     SFREQ     34
      REWIND 1                                                         SFREQ     35
      READ(1,*,END=200) OLDFILE                                        SFREQ     36
      IF ( ( OLDFILE .NE. 1 ) .AND. ( OLDFILE .NE. 2 ) ) THEN          SFREQ     37
        GOTO 200                                                       SFREQ     38
      ELSE IF ( OLDFILE .EQ. 1 ) THEN                                  SFREQ     39
*                                                                      SFREQ     40
*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME                 SFREQ     41
        IERR = 0                                                       SFREQ     42
        CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)                 SFREQ     43
        IF ( IERR .EQ. 2 ) THEN                                        SFREQ     44
          PRINT*                                                       SFREQ     45
          PRINT *, 'FILE ',PFN, ' NOT FOUND'                           SFREQ     46
          PRINT*, 'PROGRAM ABORTED!!!'                                 SFREQ     47
          PRINT*                                                       SFREQ     48
          PRINT*, 'FIND CORRECT BUILDING IDENTIFIER AND RESTART ',     SFREQ     49
     +             'PROGRAM'                                           SFREQ     50
          PRINT*                                                       SFREQ     51
          STOP                                                         SFREQ     52
```

```
*
        ELSE
          CALL LFREQ
          IF (FERR .NE. 0) CALL ERROR(5)
        END IF
      ELSE IF ( OLDFILE .EQ. 2 ) THEN
*
*** CHECK FOR EXISTENCE OF PERMANENT FILE OF SAME NAME
        IERR = 0
        CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',IERR)
        IF ( IERR .EQ. 0 ) THEN
          PRINT*
          PRINT*, 'DATA FILE ALREADY EXISTS FOR BUILDING ',BLDG
          PRINT*
          PRINT*,'IF YOU ENTER DATA AND STORE IT, YOU WILL WRITE ',
     +           'OVER THE OLD FILE.'
250       PRINT*
          PRINT*,'YOU MAY EITHER (1) ABORT OR (2) CONTINUE.'
          PRINT*,'INDICATE YOUR CHOICE BY ENTERING A NUMBER.'
          REWIND 1
          READ(1,*,END=250) ANSWER
          IF ( ANSWER .EQ. 1 ) THEN
            PRINT*
            PRINT*,'PROGRAM HAS BEEN ABORTED, AT YOUR REQUEST'
            PRINT*
            STOP
          ELSE IF ( ANSWER .EQ. 2 ) THEN
9090        CONTINUE
          ELSE
            GOTO 250
          END IF
        ELSE IF ( IERR .EQ. 2 ) THEN
*
*** NO DATA FILE ALREADY EXISTS FOR THIS BUILDING AND DATA ENTRY
*** CAN CONTINUE
9091      CONTINUE
        ELSE
*
*** PERMANENT FILE ERROR
          PRINT*
          PRINT*,'PROGRAM ABORTED !!!'
          PRINT*,'   SOME PERMANENT FILE ERROR HAS OCCURRED.'
          PRINT*,'   DOUBLE CHECK YOUR BUILDING IDENTIFICATION ',
     +           'AND TRY AGAIN'
          STOP
        END IF
*
        PRINT*
        PRINT*, ' BEGIN ENTERING DATA'
300     FTOT = FTOT + 1
        IF ( FTOT .EQ. 1) THEN
          CALL DATAIN(1,FTOT)
        ELSE
          CALL DATAIN (0,FTOT)
        END IF
400     PRINT*
        PRINT*, 'DO YOU WANT TO ENTER MORE DATA?',
     +          '(1) YES  (2) NO'
        PRINT*, '  ENTER A NUMBER !!!'
        REWIND 1
        READ(1,*,END=400) ANSWER
        IF ( (ANSWER .NE. 1) .AND. (ANSWER .NE. 2) ) THEN
          GOTO 400
        ELSE IF ( ANSWER .EQ. 1) THEN
```

```
            GOTO 300
          ELSE IF ( ANSWER .EQ. 2 ) THEN
            PRINT*
            PRINT*, 'DATA ENTRY DISCONTINUED'
          END IF
        END IF
*
*** MANIPULATE DATA
        CALL MANIP (QUIT,ABORT)
*
*** TERMINATE PROGRAM, STORING DATA IF NECESSARY
        IF ( QUIT .EQ. 1 ) THEN
         OPEN(UNIT=6,FILE=PFN(1:GETLEN(PFN)),FORM='FORMATTED',
     +           ACCESS='SEQUENTIAL',STATUS='NEW')
500       FORMAT (1PE12.6)
          DO 600 N = 1,FTOT
            WRITE (6,500) FREQA(N)
600       CONTINUE
          ENDFILE(6)
          CALL PF ('REPLACE',0,PFN(1:GETLEN(PFN)))
WARNING* NUMBER OF ARGUMENTS IN REFERENCE TO _PF IS NOT CONSISTENT
          CLOSE(6,STATUS='DELETE')
          PRINT*
          PRINT*,'DATA HAS BEEN STORED AND PROGRAM TERMINATED'
        END IF
        IF( ABORT .EQ. 1 ) THEN
          PRINT*
          PRINT*, 'PROGRAM HAS BEEN ABORTED'
          PRINT*,'      NO DATA HAS BEEN STORED !!!'
        END IF
        STOP
        END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ABORT    1031B                             INTEGER
  AFLAG       2B  /INITILN/                  REAL
  ANSWER   1032B                             INTEGER
  BLDG       0B   /INITILC/                  CHAR*5
  FERR      66B   /INITILN/                  INTEGER
  FREQ       0B   /INITILN/                  REAL
  FREQA      4B   /INITILN/                  REAL          50
  FTOT      67B   /INITILN/                  INTEGER
  IERR     1035B                             INTEGER
  LINE     NONE              UNUSED/*S*      INTEGER
  N        1034B                             INTEGER
  OLDFILE  1033B                             INTEGER
  PFN      1036B                             CHAR*7
  QUALITY    1B   /INITILN/                  INTEGER
  QUIT     1030B                             INTEGER
  RFLAG      3B   /INITILN/                  REAL
  Y1       NONE              UNUSED/*S*      INTEGER
  Y2       NONE              UNUSED/*S*      INTEGER
```

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX     INTEGER                       50


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

 DATAIN                  2       SUBROUTINE       LFREQ                    0       SUBROUTINE
 ERROR                   1       SUBROUTINE       MANIP                    2       SUBROUTINE
 GETLEN     INTEGER      1       FUNCTION         PF                       5       SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

   100     21B                     30          500     570B     FORMAT         143
   200     47B                     42          600  INACTIVE   DO-TERM         146
   250    166B                     81         9090 *NO REFS*                    92
   300    246B                    114         9091 *NO REFS*                   100
   400    260B                    120


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 SFREQ      14B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

 TAPE1   FMT/SEQ
 TAPE6   AUX/FMT/SEQ


--STATISTICS--

 PROGRAM-UNIT LENGTH            1040B =  544
 CM LABELLED COMMON LENGTH        71B =  57
 CM STORAGE USED              63000B =  26112
 COMPILE TIME                 0.260 SECONDS

      1  WARNING   ERROR IN  SFREQ
```

```
      SUBROUTINE DATAIN (INSERT,LINE)                                   SFREQ      148
***********************************************************************COMF          1
*** COMMON FOR INITIAL PARAMETERS                                   ***COMF          2
***********************************************************************COMF          3
      INTEGER FMAX                                                      COMF          4
      PARAMETER (FMAX = 50)                                             COMF          5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF          6
     $                 FTOT                                             COMF          7
      COMMON /INITILC/ BLDG                                             COMF          8
      CHARACTER * 5 BLDG                                                COMF          9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF         10
      INTEGER QUALITY, FERR, FTOT                                       COMF         11
***********************************************************************COMF         12
***********************************************************************COMF         13
      INTEGER INSERT,LINE                                               SFREQ      150
      IF(INSERT.EQ.1) THEN                                              SFREQ      151
 200  PRINT *, ' ENTER FREQUENCY FOR LINE #',LINE                       SFREQ      152
300   READ(1,*,END=200,ERR=200) FREQA(LINE)                             SFREQ      153
      ENDIF                                                             SFREQ      154
      IF(INSERT.EQ.0) THEN                                              SFREQ      155
 400  PRINT *, 'ENTER NEXT FREQS, ONE PER LINE AFTER' ,                 SFREQ      156
     + ' EACH QUESTION MARK.'                                           SFREQ      157
      PRINT *, ' ENTER ZERO (0.0) TO DISCONTINUE ENTRIES'               SFREQ      158
      PRINT *, 'START WITH LINE NUMBER = ', LINE                        SFREQ      159
  500 REWIND 1                                                          SFREQ      160
      READ(1,*,END=400,ERR=400) FREQA(LINE)                             SFREQ      161
      IF(FREQA(LINE).GT.0.0) THEN                                       SFREQ      162
*******************************************************                 SFREQ      163
*  CHECK IF ARRAY SIZE EXCEEDED                                         SFREQ      164
*******************************************************                 SFREQ      165
      IF( LINE.GT.FMAX) THEN                                            SFREQ      166
      PRINT *, 'MAXIMUM NUMBER OF DATA LINES CANNOT '                   SFREQ      167
      PRINT *, 'EXCEED ', FMAX, '. INSERTION NOT POSSIBLE.'             SFREQ      168
      RETURN                                                            SFREQ      169
      ENDIF                                                             SFREQ      170
*******************************************************                 SFREQ      171
      LINE = LINE + 1                                                   SFREQ      172
      GOTO 500                                                          SFREQ      173
      ELSE                                                              SFREQ      174
      LINE =LINE -1                                                     SFREQ      175
      ENDIF                                                             SFREQ      176
      ENDIF                                                             SFREQ      177
      RETURN                                                            SFREQ      178
      END                                                               SFREQ      179
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
AFLAG	2B	/INITILN/		REAL	
BLDG	0B	/INITILC/		CHAR*5	
FERR	66B	/INITILN/		INTEGER	
FREQ	0B	/INITILN/		REAL	
FREQA	4B	/INITILN/		REAL	50
FTOT	67B	/INITILN/		INTEGER	
INSERT	1	DUMMY-ARG		INTEGER	
LINE	2	DUMMY-ARG		INTEGER	
QUALITY	1B	/INITILN/		INTEGER	
RFLAG	3B	/INITILN/		REAL	

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-----------------------VALUE

  FMAX     INTEGER                      50


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    200     12B                  17
    300 *NO REFS*                18
    400     25B                  21
    500     33B                  25


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DATAIN      5B     2


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH            200B =  128
  CM LABELLED COMMON LENGTH       71B =  57
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.073 SECONDS
```

```
      SUBROUTINE MANIP (QUIT,ABORT)                                     SFREQ    180
************************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF       2
************************************************************************COMF       3
      INTEGER FMAX                                                      COMF       4
      PARAMETER (FMAX = 50)                                             COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF       6
     $                 FTOT                                             COMF       7
      COMMON /INITILC/ BLDG                                             COMF       8
      CHARACTER * 5 BLDG                                                COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF      10
      INTEGER QUALITY, FERR, FTOT                                       COMF      11
************************************************************************COMF      12
************************************************************************COMF      13
      INTEGER ABORT,ANSWER,DOK,FLAG1,LOK,N,NOK,OK,OK1,OK2,QUIT,INSERT   SFREQ    182
      INTEGER TEMP,V,X,Y,COMMAND                                        SFREQ    183
      CHARACTER * 3 DIR, FROM, TO                                       SFREQ    184
*                                                                       SFREQ    185
   10 FLAG1 = 0                                                         SFREQ    186
      PRINT*                                                            SFREQ    187
      PRINT*, 'CHOOSE'                                                  SFREQ    188
      PRINT*, '   (1) DISPLAY LINE OF DATA  (4) DISPLAY ALL LINES'      SFREQ    189
      PRINT*, '   (2) INSERT LINE INTO FILE (5) APPEND LINES OF DATA'   SFREQ    190
      PRINT*, '   (3) DELETE LINE           (6) STORE DATA AND EXIT ',  SFREQ    191
     +        'PROGRAM'                                                 SFREQ    192
      PRINT*, '                             (7) EXIT PROGRAM WITHOUT ', SFREQ    193
     +        'STORING DATA'                                            SFREQ    194
      PRINT*, 'ENTER A NUMBER !!!'                                      SFREQ    195
      PRINT*                                                            SFREQ    196
      REWIND 1                                                          SFREQ    197
      READ(1,*,END=10) COMMAND                                          SFREQ    198
*                                                                       SFREQ    199
*---------------------------------------------------------------------- SFREQ    200
*** DISPLAY LINE ***                                                    SFREQ    201
*---------------------------------------------------------------------- SFREQ    202
      IF ( COMMAND .EQ. 1 ) THEN                                        SFREQ    203
*                                                                       SFREQ    204
*** INDICATE EMPTY DATA FILE                                            SFREQ    205
        IF ( FTOT .EQ. 0 ) THEN                                         SFREQ    206
          PRINT*                                                        SFREQ    207
          PRINT*, 'DATA FILE IS EMPTY !!!'                              SFREQ    208
*                                                                       SFREQ    209
*** ENTER NUMBER OF LINE TO BE DISPLAYED                                SFREQ    210
        ELSE                                                            SFREQ    211
  100     PRINT*                                                        SFREQ    212
          PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DISPLAYED'      SFREQ    213
          PRINT*, '     ( ENTER "0" TO ESCAPE DISPLAY MODE )'           SFREQ    214
          REWIND 1                                                      SFREQ    215
          READ(1,*,END=100) N                                           SFREQ    216
*                                                                       SFREQ    217
*** CHECK VALIDITY OF LINE NUMBER                                       SFREQ    218
          IF ( (N .GT. FTOT) .OR. (N .LT. 0) ) THEN                     SFREQ    219
            PRINT*                                                      SFREQ    220
            PRINT*, 'INCORRECT NUMBER !!!!!!      TRY AGAIN !!!'        SFREQ    221
            PRINT*, '            -OR-  ENTER "0" TO ESCAPE FROM ',      SFREQ    222
     +              '"DISPLAY" MODE'                                    SFREQ    223
            GOTO 100                                                    SFREQ    224
*                                                                       SFREQ    225
*** ABORT 'DISPLAY' MODE                                                SFREQ    226
          ELSE IF ( N .EQ. 0 ) THEN                                     SFREQ    227
            PRINT*                                                      SFREQ    228
            PRINT*, '    "DISPLAY" MODE ABORTED !!!'                    SFREQ    229
*                                                                       SFREQ    230
*** DISPLAY LINE OF DATA                                                SFREQ    231
```

```
           ELSE IF ( (N .GT. 0) .AND. (N .LE. FTOT) ) THEN
             PRINT*
             CALL DISPLAY( N, COMMAND)
*
           END IF
         END IF
       END IF
*
*-------------------------------------------------------------------
*** INSERT LINE ***
*-------------------------------------------------------------------
       IF ( COMMAND .EQ. 2 ) THEN
*
*** INDICATE EMPTY DATA FILE
         IF ( FTOT .EO. 0 ) THEN
           PRINT*
           PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** REOUEST NUMBER OF LINE BEFORE WHICH INSERTION IS TO BE MADE
         ELSE
 200       PRINT*
           PRINT*, 'SPECIFY NUMBER OF LINE BEFORE WHICH A NEW LINE IS ',
     +             'TO BE INSERTED'
           PRINT*, '     ( ENTER "0" TO ESCAPE "INSERTION" MODE )'
           REWIND 1
           READ(1,*,END=200) N
*
*** CHECK FOR VALID LINE NUMBER
           IF ( ( N .LT. 0 ) .OR. ( N .GT. FTOT ) ) THEN
             PRINT*
             PRINT*, 'INCORRECT LINE NUMBER !!!'
             PRINT*, '      TRY AGAIN !!!  -OR-  ENTER "0" TO ESCAPE',
     +               '"INSERTION" MODE'
             GOTO 200
*
*** ABORT INSERTION MODE
           ELSE IF ( N .EQ. 0 ) THEN
             PRINT*
             PRINT*, '       "INSERTION" MODE ABORTED'
*
*** MAKE ROOM FOR NEW LINE OF DATA
           ELSE IF ( (N .GT. 0) .AND. (N .LE. FTOT) ) THEN
*********************************************************
*  CHECK IF ARRAY SIZE EXCEEDED
*********************************************************
       IF( FTOT.EO.FMAX) THEN
       PRINT *, 'MAXIMUM NUMBER OF DATA LINES CANNOT '
       PRINT *, 'EXCEED ', FMAX, '. INSERTION NOT POSSIBLE.'
       GO TO 10
       ENDIF
*********************************************************
             DO 230 X = FTOT,N,-1
                 FREQA(X+1) = FREOA(X)
 210           CONTINUE
 230         CONTINUE
*
*** ENTER DATA FOR NEW LINE
             FTOT = FTOT + 1
             CALL DATAIN (1,N)
*
               PRINT*
               PRINT*, 'THE FOLLOWING LINE HAS BEEN ADDED AS LINE ', N
               CALL DISPLAY( N, COMMAND)
           END IF
```

```
          END IF                                                     SFREQ
        END IF                                                       SFREQ
*                                                                    SFREQ
*------------------------------------------------------------------- SFREQ
*** DELETE LINE ***                                                  SFREQ
*------------------------------------------------------------------- SFREQ
        IF ( COMMAND .EQ. 3 ) THEN                                   SFREQ
*                                                                    SFREQ
*** INDICATE EMPTY DATA FILE                                         SFREQ
          IF ( FTOT .EQ. 0 ) THEN                                    SFREQ
            PRINT*                                                   SFREQ
            PRINT*, 'DATA FILE IS EMPTY !!!'                         SFREQ
*                                                                    SFREQ
*** READ NUMBER OF LINE TO BE DELETED                                SFREQ
          ELSE                                                       SFREQ
300         PRINT*                                                   SFREQ
            PRINT*, 'SPECIFY THE NUMBER OF THE LINE TO BE DELETED'   SFREQ
            PRINT*, '   (ENTER "0" TO ESCAPE DELETION MODE)'         SFREQ
            REWIND 1                                                 SFREQ
            READ(1,*,END=300) N                                      SFREQ
*                                                                    SFREQ
*** CHECK VALIDITY OF LINE NUMBER                                    SFREQ
            IF ( (N .GT. FTOT ) .OR. ( N .LT. 0 ) ) THEN             SFREQ
              PRINT*                                                 SFREQ
              PRINT*, ' INCORRECT NUMBER !!!'                        SFREQ
              PRINT*, '    TRY AGAIN !!!  -OR-  ENTER "0" TO ESCAPE FROM',SFREQ
     +              '"DELETE" MODE'                                  SFREQ
              GOTO 300                                               SFREQ
*                                                                    SFREQ
*** ABORT 'DELETE' MODE                                              SFREQ
            ELSE IF ( N .EQ. 0 ) THEN                                SFREQ
              PRINT*, '  "DELETE" MODE ABORTED'                      SFREQ
*                                                                    SFREQ
*** DOUBLE CHECK CHOICE OF LINE TO BE DELETED                        SFREQ
            ELSE IF (( N .GT. 0 ) .AND. ( N .LE. FTOT )) THEN        SFREQ
              PRINT*                                                 SFREQ
              PRINT*, 'DOUBLE CHECK !!!'                             SFREQ
              PRINT*, '     DO YOU WANT TO DELETE THE FOLLOWING LINE?:'  SFREQ
              CALL DISPLAY( N, COMMAND)                              SFREQ
305           PRINT*, '   ENTER (1) YES  OR   (2) NO'                SFREQ
              REWIND 1                                               SFREQ
              READ(1,*,END=305) ANSWER                               SFREQ
*                                                                    SFREQ
*** DELETE LINE                                                      SFREQ
              IF ( ANSWER .EQ. 1 ) THEN                              SFREQ
                DO 330 X = N, FTOT - 1                               SFREQ
                   FREQA(X) = FREQA(X+1)                             SFREQ
330             CONTINUE                                             SFREQ
                FTOT = FTOT - 1                                      SFREQ
                PRINT*                                               SFREQ
                PRINT*, 'LINE # ',N,' DELETED'                       SFREQ
              END IF                                                 SFREQ
*                                                                    SFREQ
            END IF                                                   SFREQ
          END IF                                                     SFREQ
        END IF                                                       SFREQ
*                                                                    SFREQ
*------------------------------------------------------------------- SFREQ
*** DISPLAY ALL DATA ***                                             SFREQ
*------------------------------------------------------------------- SFREQ
        IF ( COMMAND .EQ. 4 ) THEN                                   SFREQ
*                                                                    SFREQ
*** INDICATE EMPTY DATA FILE                                         SFREQ
          IF ( FTOT .EQ. 0 ) THEN                                    SFREQ
```

```
          PRINT*
          PRINT*, 'DATA FILE IS EMPTY !!!'
*
*** DISPLAY DATA
        ELSE
          PRINT*
          CALL DISPLAY( N, COMMAND)
*
        END IF
      END IF
*
*-------------------------------------------------------------------
*** ADD DATA ***
*-------------------------------------------------------------------
      IF ( COMMAND .EQ. 5 ) THEN
*
*** ENTER DATA
500     FTOT = FTOT + 1
*************************************************************
*  CHECK IF ARRAY SIZE EXCEEDED
*************************************************************
      IF( FTOT.EQ.FMAX) THEN
      PRINT *, 'MAXIMUM NUMBER OF DATA LINES CANNOT '
      PRINT *, 'EXCEED ', FMAX, '. INSERTION NOT POSSIBLE.'
      GO TO 10
      ENDIF
*************************************************************
        CALL DATAIN (0,FTOT)
510     PRINT*
        PRINT*, 'DO YOU WANT TO ENTER MORE DATA?   (1) YES  (2) NO'
        PRINT*, '    ENTER A NUMBER !!!'
        REWIND 1
        READ(1,*,END=510) ANSWER
*
*** CHECK VALIDITY OF NUMBER
        IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN
          GOTO 510
*
*** ENTER MORE DATA
        ELSE IF ( ANSWER .EQ. 1 ) THEN
          GOTO 500
*
*** DISCONTINUE DATA ENTRY
        ELSE IF ( ANSWER .EQ. 2 ) THEN
          PRINT*
          PRINT*, 'DATA ENTRY DISCONTINUED'
*
        END IF
      END IF
*
*-------------------------------------------------------------------
*** STORE DATA AND PROGRAM ***
*-------------------------------------------------------------------
      IF ( COMMAND .EQ. 6 ) THEN
 600    PRINT*
        PRINT*, 'DOUBLE CHECK !!!'
        PRINT*, '     DO YOU YOU WANT TO STORE THIS DATA AND END PROG'
        PRINT*, '  NOTE: STORING THIS DATA WILL WIPE OUT ANY OLD FILE '
        PRINT*, '  OF THE SAME NAME !!!'
        PRINT*, '         ENTER A NUMBER: (1) YES   (2) NO'
        REWIND 1
        READ(1,*,END=600) ANSWER
*
*** SET FLAG FOR STORING DATA IN THE MAIN PROGRAM
```

```
          IF ( ANSWER .EQ. 1 ) THEN
            QUIT = 1
            RETURN
*
*** ABORT 'STORING' MODE
          ELSE IF ( ANSWER .EQ. 2 ) THEN
            PRINT*
            PRINT*, '    "STORING" MODE DISCONTINUED'
*
*** CHECK VALIDITY OF ANSWER
          ELSE IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN
            GOTO 600
*
          END IF
        END IF
*
*-----------------------------------------------------------------------
*** END PROGRAM WITHOUT STORING DATA ***
*-----------------------------------------------------------------------
        IF ( COMMAND .EQ. 7 ) THEN
  700     PRINT*
          PRINT*, 'DOUBLE CHECK !!!'
          PRINT*, '    DO YOU WANT TO END THIS PROGRAM ',
     +        'WITHOUT STORING DATA?'
          PRINT*, '        ENTER A NUMBER: (1) YES   (2) NO'
          REWIND 1
          READ(1,*,END=700) ANSWER
*
*** SET FLAG FOR ABORTING PROGRAM IN THE MAIN PROGRAM
          IF ( ANSWER .EQ. 1 ) THEN
            ABORT = 1
            RETURN
*
*** ABORT 'STORING' MODE
          ELSE IF ( ANSWER .EQ. 2 ) THEN
            PRINT*
            PRINT*, '    "ABORTION" MODE DISCONTINUED'
*
*** CHECK VALIDITY OF ANSWER
          ELSE IF ( ( ANSWER .NE. 1 ) .AND. ( ANSWER .NE. 2 ) ) THEN
            GOTO 700
*
          END IF
        END IF
*
*-----------------------------------------------------------------------
*** LOOP TO BEGINNING OF 'MANIP' SUBROUTINE
*-----------------------------------------------------------------------
        GOTO 10
*
        END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ABORT       2   DUMMY-ARG                   INTEGER
  AFLAG       2B  /INITILN/                   REAL
  ANSWER   1364B                              INTEGER
  BLDG        0B  /INITILC/                   CHAR*5
  COMMAND  1370B                              INTEGER
  DIR      NONE              UNUSED/*S*       CHAR*3
  DOK      NONE              UNUSED/*S*       INTEGER
  FERR       66B  /INITILN/                   INTEGER
```

```
 FLAG1     1365B                                  INTEGER
 FREQ         0B  /INITILN/                       REAL
 FREQA        4B  /INITILN/                       REAL           50
 FROM      NONE                 UNUSED/*S*        CHAR*3
 FTOT        67B  /INITILN/                       INTEGER
 INSERT    NONE                 UNUSED/*S*        INTEGER
 LOK       NONE                 UNUSED/*S*        INTEGER
 N         1366B                                  INTEGER
 NOK       NONE                 UNUSED/*S*        INTEGER
 OK        NONE                 UNUSED/*S*        INTEGER
 OK1       NONE                 UNUSED/*S*        INTEGER
 OK2       NONE                 UNUSED/*S*        INTEGER
 QUALITY      1B  /INITILN/                       INTEGER
 QUIT         1   DUMMY-ARG                       INTEGER
 RFLAG        3B  /INITILN/                       REAL
 TEMP      NONE                 UNUSED/*S*        INTEGER
 TO        NONE                 UNUSED/*S*        CHAR*3
 V         NONE                 UNUSED/*S*        INTEGER
 X         1367B                                  INTEGER
 Y         NONE                 UNUSED/*S*        INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-----------------------VALUE

  FMAX    INTEGER                        50


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  DATAIN                  2      SUBROUTINE
  DISPLAY                 2      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    10      7B                     19          305    325B                    168
   100     50B                     45          330  INACTIVE   DO-TERM        176
   200    133B                     85          500    413B                    210
   210 *NO REFS*                  118          510    427B                    221
   230  INACTIVE   DO-TERM        119          600    470B                    247
   300    256B                    144          700    542B                    277


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  MANIP       5B     2


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ
```



```
--STATISTICS--

  PROGRAM-UNIT LENGTH             1375B =   765
  CM LABELLED COMMON LENGTH         71B =    57
  CM STORAGE USED               63000B = 26112
  COMPILE TIME                  0.406 SECONDS
```

```
FTN 5.1+552          83/12/20. 11.52.59 PAGE     14
SUBROUTINE DISPLAY   74/175   OPT=0

      1        SUBROUTINE DISPLAY (LINE, COMMAND)                                  SFREQ      475
      2 *************************************************************************COMF         1
      3 *** COMMON FOR INITIAL PARAMETERS                                      ***COMF         2
      4 *************************************************************************COMF         3
      5        INTEGER FMAX                                                        COMF         4
      6        PARAMETER (FMAX = 50)                                               COMF         5
      7        COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,    COMF         6
      8      $                  FTOT                                               COMF         7
      9        COMMON /INITILC/ BLDG                                               COMF         8
     10        CHARACTER * 5 BLDG                                                  COMF         9
     11        REAL FREQ, AFLAG, RFLAG, FREQA                                      COMF        10
     12        INTEGER QUALITY, FERR, FTOT                                         COMF        11
     13 *************************************************************************COMF        12
     14 *************************************************************************COMF        13
     15        INTEGER LINE, COMMAND, N                                            SFREQ      477
     16 1000   FORMAT (1X,'LINE #      FREQUENCY (HZ)')                            SFREQ      478
     17 2000   FORMAT (4X,I3,8X,1PE15.5)                                           SFREQ      479
     18        PRINT 1000                                                          SFREQ      480
     19        IF ( COMMAND .EQ. 4 ) THEN                                          SFREQ      481
     20          DO 10 N = 1,FTOT                                                  SFREQ      482
     21            PRINT 2000, N,FREQA(N)                                          SFREQ      483
     22 10       CONTINUE                                                          SFREQ      484
     23        ELSE                                                                SFREQ      485
     24            PRINT 2000, LINE,FREQA(LINE)                                    SFREQ      486
     25        END IF                                                              SFREQ      487
     26        RETURN                                                              SFREQ      488
     27        END                                                                 SFREQ      489


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG       2B   /INITILN/                 REAL
  BLDG        0B   /INITILC/                 CHAR*5
  COMMAND     2    DUMMY-ARG                 INTEGER
  FERR       66B   /INITILN/                 INTEGER
  FREQ        0B   /INITILN/                 REAL
  FREQA       4B   /INITILN/                 REAL            50
  FTOT       67B   /INITILN/                 INTEGER
  LINE        1    DUMMY-ARG                 INTEGER
  N         100B                             INTEGER
  QUALITY     1B   /INITILN/                 INTEGER
  RFLAG       3B   /INITILN/                 REAL


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-----------------------VALUE

  FMAX     INTEGER                         50


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        22
  1000      51B    FORMAT         16
  2000      56B    FORMAT         17
```

```
FTN 5.1+552          83/12/20. 11.52.59 PAGE     15
SUBROUTINE DISPLAY   74/175   OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DISPLAY      5B      2


--STATISTICS--

  PROGRAM-UNIT LENGTH              104B =  68
  CM LABELLED COMMON LENGTH         71B =  57
  CM STORAGE USED                61000B =  25088
  COMPILE TIME                   0.045 SECONDS
```

```
FTN 5 1+552        83/12/20  11.52.59 PAGE    16
  FUNCTION VAL       74/175  OPT=0

      INTEGER FUNCTION VAL(STRING)                                    SFREQ     490
C** RETURNS THE INTEGER VALUE OF A STRING.                            SFREQ     491
      INTEGER NUMBER, X,L,EXP,DIGIT,GETLEN                            SFREQ     492
      CHARACTER * (*) STRING                                          SFREQ     493
      L = GETLEN(STRING)                                              SFREQ     494
      NUMBER = 0                                                      SFREQ     495
      DO 10 X = L,1,-1                                                SFREQ     496
      EXP = L - X                                                     SFREQ     497
      DIGIT = ICHAR(STRING(X:X)) - 16                                 SFREQ     498
      NUMBER = NUMBER + DIGIT*10**EXP                                 SFREQ     499
 10   CONTINUE                                                        SFREQ     500
      VAL = NUMBER                                                    SFREQ     501
      RETURN                                                          SFREQ     502
      END                                                             SFREQ     503


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  DIGIT      76B                                 INTEGER
  EXP        75B                                 INTEGER
  L          74B                                 INTEGER
  NUMBER     72B                                 INTEGER
  STRING      1   DUMMY-ARG                      CHAR*(*)
  VAL        71B                                 INTEGER
  X          73B                                 INTEGER


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN    INTEGER        1      FUNCTION
  ICHAR     INTEGER        1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        11


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  VAL         6B     1


--STATISTICS--

  PROGRAM-UNIT LENGTH           102B =  66
  CM STORAGE USED             61000B =  25088
  COMPILE TIME                0.039 SECONDS
```

```
      INTEGER FUNCTION GETLEN (STRING)                                 SFREQ   504
C                                                                      SFREQ   505
C DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING               SFREQ   506
C                                                                      SFREQ   507
C                                                                      SFREQ   508
C ARGUMENT DEFINITIONS --                                              SFREQ   509
C   READ ARGUMENTS                                                     SFREQ   510
C     STRING - STRING WHOSE LENGTH IS TO BE DETERMINED                 SFREQ   511
C                                                                      SFREQ   512
      CHARACTER * (*) STRING                                           SFREQ   513
C                                                                      SFREQ   514
C FUNCTION PARAMETERS                                                  SFREQ   515
      CHARACTER * 1 BLANK                                              SFREQ   516
      PARAMETER (BLANK = ' ')                                          SFREQ   517
C                                                                      SFREQ   518
C LOCAL VARIABLES                                                      SFREQ   519
      INTEGER NEXT                                                     SFREQ   520
C                                                                      SFREQ   521
C START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK           SFREQ   522
      DO 10 NEXT = LEN(STRING),1,-1                                    SFREQ   523
         IF (STRING(NEXT : NEXT) .NE. BLANK) THEN                      SFREQ   524
            GETLEN = NEXT                                              SFREQ   525
            RETURN                                                     SFREQ   526
         END IF                                                        SFREQ   527
   10 CONTINUE                                                         SFREQ   528
C                                                                      SFREQ   529
C ALL CHARACTERS ARE BLANKS                                            SFREQ   530
      GETLEN = 0                                                       SFREQ   531
C                                                                      SFREQ   532
      RETURN                                                           SFREQ   533
      END                                                              SFREQ   534
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 GETLEN     63B                             INTEGER
 NEXT       64B                             INTEGER
 STRING      1   DUMMY-ARG                  CHAR*(*)


--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE-----------------------VALUE

 BLANK   CHAR*1                       ' '


--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 LEN       INTEGER        1      INTRINSIC


--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        25
```

```
FTN 5.1+552          83/12/20. 11.52.59 PAGE     18
  FUNCTION GETLEN    74/175  OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  GETLEN      6B      1


--STATISTICS--

  PROGRAM-UNIT LENGTH              70B =   56
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.039 SECONDS
```

```
      SUBROUTINE LFREQ                                                  LFREQ      1
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LFREQ      2
*!!!                                                                 !!!LFREQ      3
*!!!   LOAD THE CONTENTS OF THE FILE 'BXXXXXF' INTO ARRAYS FREQA.       LFREQ      4
*!!!                                                                 !!!LFREQ      5
*!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!LFREQ      6
************************************************************************LFREQ      7
************************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF       2
************************************************************************COMF       3
      INTEGER FMAX                                                      COMF       4
      PARAMETER (FMAX = 50)                                             COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF       6
     $                  FTOT                                            COMF       7
      COMMON /INITILC/ BLDG                                             COMF       8
      CHARACTER * 5 BLDG                                                COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF      10
      INTEGER QUALITY, FERR, FTOT                                       COMF      11
************************************************************************COMF      12
************************************************************************COMF      13
***************************************************                     LFREQ      9
*   DECLARATION OF VARIABLES                                            LFREQ     10
***************************************************                     LFREQ     11
      INTEGER GETLEN, R, C                                              LFREQ     12
      CHARACTER * 7 NAME, PFN                                           LFREQ     13
***************************************************                     LFREQ     14
*                                                                       LFREQ     15
***************************************************                     LFREQ     16
      NAME = 'B'//BLDG(1:GETLEN(BLDG))//'F'                             LFREQ     17
      PFN = NAME (1:GETLEN(NAME))                                       LFREQ     18
      FERR = 0                                                          LFREQ     19
      CALL PF ('GET',0,PFN(1:GETLEN(PFN)),'RC',FERR)                    LFREQ     20
      IF ( FERR .EQ. 0 ) THEN                                           LFREQ     21
        OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',                       LFREQ     22
     $        STATUS='OLD', ACCESS='SEQUENTIAL')                        LFREQ     23
        FTOT = 0                                                        LFREQ     24
        DO 10 R = 1,FMAX                                                LFREQ     25
          READ (3,1000,END=20) FREQA(R)                                 LFREQ     26
 1000     FORMAT(E12.7)                                                 LFREQ     27
          FTOT = FTOT + 1                                               LFREQ     28
10      CONTINUE                                                        LFREQ     29
20      CONTINUE                                                        LFREQ     30
        CLOSE(3,STATUS='DELETE')                                        LFREQ     31
      ELSE IF ( FERR .EQ. 2 ) THEN                                      LFREQ     32
        CALL WARNING (11)                                               LFREQ     33
      ELSE                                                              LFREQ     34
        CALL WARNING (12)                                               LFREQ     35
      END IF                                                            LFREQ     36
      RETURN                                                            LFREQ     37
      END                                                               LFREQ     38
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
AFLAG	2B	/INITILN/		REAL	
BLDG	0B	/INITILC/		CHAR*5	
C	NONE		UNUSED/*S*	INTEGER	
FERR	66B	/INITILN/		INTEGER	
FREQ	0B	/INITILN/		REAL	
FREQA	4B	/INITILN/		REAL	50
FTOT	67B	/INITILN/		INTEGER	
NAME	210B			CHAR*7	
PFN	211B			CHAR*7	

```
 QUALITY      1B  /INITILN/                        INTEGER
 R          207B                                   INTEGER
 RFLAG        3B  /INITILN/                        REAL


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX    INTEGER                        50


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 GETLEN    INTEGER         1      FUNCTION
 PF                        5      SUBROUTINE
 WARNING                   1      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        41
    20     63B                    42
  1000    120B     FORMAT         39


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 LFREQ       5B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

 TAPE3   AUX/FMT/SEQ


--STATISTICS--

 PROGRAM-UNIT LENGTH            215B =  141
 CM LABELLED COMMON LENGTH       71B =  57
 CM STORAGE USED              61000B =  25088
 COMPILE TIME                 0.080 SECONDS
```

```
      SUBROUTINE ERROR(IERR)                                             ERROR       1
      CHARACTER*45 MESSAGE(20)                                           ERROR       2
      DATA MESSAGE( 1)/'MATERIALS DATA BASE IS EMPTY              '/    ERROR       3
      DATA MESSAGE( 2)/'FREQUENCY IS OUT OF RANGE                 '/    ERROR       4
      DATA MESSAGE( 3)/'THIS MATERIAL IS NOT IN DATA BASE         '/    ERROR       5
      DATA MESSAGE( 4)/'DENOMINATOR IS ZERO                       '/    ERROR       6
      DATA MESSAGE( 5)/'FILE HANDLING ERROR                       '/    ERROR       7
      DATA MESSAGE( 6)/'ERROR CODE IS OUT OF RANGE                '/    ERROR       8
      DATA MESSAGE( 7)/'ERROR CODE IS OUT OF RANGE                '/    ERROR       9
      DATA MESSAGE( 8)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      10
      DATA MESSAGE( 9)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      11
      DATA MESSAGE(10)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      12
      DATA MESSAGE(11)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      13
      DATA MESSAGE(12)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      14
      DATA MESSAGE(13)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      15
      DATA MESSAGE(14)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      16
      DATA MESSAGE(15)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      17
      DATA MESSAGE(16)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      18
      DATA MESSAGE(17)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      19
      DATA MESSAGE(18)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      20
      DATA MESSAGE(19)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      21
      DATA MESSAGE(20)/'ERROR CODE IS OUT OF RANGE                '/    ERROR      22
      IERRM=5                                                            ERROR      23
      IF(IERR.GT.IERRM) IERR=20                                          ERROR      24
      WRITE(6,10) IERR,MESSAGE(IERR)                                     ERROR      25
10    FORMAT(' ***ERROR NUMBER = ',I5,' ***  ',A45)                      ERROR      26
      CALL PMDSTOP                                                       ERROR      27
      STOP 'ERROR'                                                       ERROR      28
      END                                                                ERROR      29
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  IERR        1   DUMMY-ARG                  INTEGER
  IERRM     210B                             INTEGER
  MESSAGE    56B                             CHAR*45         20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  PMDSTOP                  0      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10     36B     FORMAT          26


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ERROR       5B     1
```



```
--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE6   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH             213B =  139
  CM STORAGE USED               61000B =  25088
  COMPILE TIME                  0.056 SECONDS
```

```
      SUBROUTINE WARNING(ERR)                                         WARNING
      INTEGER ERR, ERRM                                               WARNING
      CHARACTER*45 MESSAGE(20)                                        WARNING
      DATA MESSAGE( 1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING
      DATA MESSAGE( 2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE    '/ WARNING
      DATA MESSAGE( 3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG   '/ WARNING
      DATA MESSAGE( 4)/'FILE HANDLING PROBLEM ON "MATTER FILE        '/ WARNING
      DATA MESSAGE( 5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING
      DATA MESSAGE( 6)/'FILE HANDLING PROBLEM ON "TYPE" FILE         '/ WARNING
      DATA MESSAGE( 7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING
      DATA MESSAGE( 8)/'FILE HANDLING PROBLEM ON "WALL" FILE         '/ WARNING
      DATA MESSAGE( 9)/'HEIGHT AND WIDTH OF ROOM MISSING             '/ WARNING
      DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING                    '/ WARNING
      DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG       '/ WARNING
      DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE         '/ WARNING
      DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING
      DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING
      DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING
      DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING
      DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING
      DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING
      DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING
      DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING
      ERRM=12                                                         WARNING
      IERR = ERR                                                      WARNING
      IF(ERR.GT.ERRM) IERR=20                                         WARNING
      WRITE(6,20)                                                     WARNING
      WRITE(6,10) ERR,MESSAGE(IERR)                                   WARNING
      WRITE(6,20)                                                     WARNING
10    FORMAT(' ***WARNING NUMBER = ',I5,' ***  ',A45)                 WARNING
20    FORMAT('      ')                                                WARNING
      RETURN                                                          WARNING
      END                                                             WARNING
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 ERR          1   DUMMY-ARG                  INTEGER
 ERRM       60B                              INTEGER
 IERR      213B                              INTEGER
 MESSAGE    61B                              CHAR*45        20

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10      34B     FORMAT          30
    20      42B     FORMAT          31

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 WARNING      5B      1
```

```
FTN 5.1+552          83/12/20. 11.52.59 PAGE    24
SUBROUTINE WARNING  74/175  OPT=0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

 TAPE6   FMT/SEQ

--STATISTICS--

 PROGRAM-UNIT LENGTH             216B =  142
 CM STORAGE USED               61000B =  25088
 COMPILE TIME                  0.064 SECONDS
```

Appendix 9.7 Listing of Computer Program MASTER

```
      PROGRAM MASTER (INPUT,OUTPUT, TAPE1 = INPUT,TAPE6 = OUTPUT)       MASTER   1
************************************************************************MASTER   2
************************************************************************COMF     1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF     2
************************************************************************COMF     3
      INTEGER FMAX                                                      COMF     4
      PARAMETER (FMAX = 50)                                             COMF     5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF     6
     $                 FTOT                                             COMF     7
      COMMON /INITILC/ BLDG                                             COMF     8
      CHARACTER * 5 BLDG                                                COMF     9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF    10
      INTEGER QUALITY, FERR, FTOT                                       COMF    11
************************************************************************COMF    12
************************************************************************COMF    13
************************************************************************COMW     1
***   COMMON FOR DATABASE OF WALL PARAMETERS                         ***COMW     2
************************************************************************COMW     3
      INTEGER WMAX                                                      COMW     4
      PARAMETER (WMAX = 75)                                             COMW     5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW     6
      COMMON /WALLC/ WALL(WMAX,4)                                       COMW     7
      INTEGER WTOT,WERR                                                 COMW     8
      REAL WDIM                                                         COMW     9
      CHARACTER *3 WALL                                                 COMW    10
* =================================                                     COMW    11
** DESCRIPTION OF ARRAYS                                                COMW    12
* =================================                                     COMW    13
*    WALL IDENTIFICATION                                                COMW    14
* ---------------------------------                                     COMW    15
* DIRECTION    FROM       TO                                            COMW    16
*              ROOM       ROOM                                          COMW    17
* ---------------------------------                                     COMW    18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                     COMW    19
*  A3          A3          A3                                           COMW    20
* ======================================================                COMW    21
*        WALL PARAMETERS                                                COMW    22
* ------------------------------------------------------                COMW    23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                   COMW    24
* ------------------------------------------------------                COMW    25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                         COMW    26
*  A3         F8.2        F8.2        F8.2                              COMW    27
************************************************************************COMW    28
************************************************************************COMW    29
************************************************************************COMT     1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT     2
************************************************************************COMT     3
      INTEGER TMAX                                                      COMT     4
      PARAMETER (TMAX=35)                                               COMT     5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT     6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT     7
      INTEGER TTOT,TDBTOT,TERR                                          COMT     8
      REAL TDIM,TDB2                                                    COMT     9
      CHARACTER * 3 TYPE,TDB1                                           COMT    10
*==============================================                         COMT    11
* DESCRIPTION OF ARRAYS                                                 COMT    12
*==============================================                         COMT    13
*   ID        MATERIAL    FRAME MATERIAL                                COMT    14
*----------------------------------------------                         COMT    15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                     COMT    16
* A3          A3          A3                                            COMT    17
*==============================================                         COMT    18
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT    19
*                         THICKNESS   ABOVE FLOOR                       COMT    20
```

```
*-----------------------------------------------------                  COMT      21
* TDIM(X,1)    TDIM(X,2)    TDIM(X,3)    TDIM(X,4)                     COMT      22
*  F8.2         F8.2         F8.2         F8.2                         COMT      23
*=====================================================                 COMT      24
*  ID       ATTENUATION    AREA                                        COMT      25
* ----------------------------------------------------                 COMT      26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                    COMT      27
*   A3          E9.3         E9.3                                      COMT      28
************************************************************************COMT      29
************************************************************************COMT      30
************************************************************************COMH       1
***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH       2
************************************************************************COMH       3
      INTEGER HMAX                                                      COMH       4
      PARAMETER (HMAX = 35)                                             COMH       5
      COMMON /HOLEN/ HTOT, HERR                                         COMH       6
      COMMON /HOLEC/ HOLE(HMAX,4)                                       COMH       7
      INTEGER HTOT, HERR                                                COMH       8
      CHARACTER * 3 HOLE                                                COMH       9
* ====================================================                  COMH      10
* DESCRIPTION OF ARRAYS                                                 COMH      11
* ====================================================                  COMH      12
*      ROOM IDENTIFICATION                  APERTURE ID                 COMH      13
* -----------------------------------                                   COMH      14
* DIRECTION    FROM ROOM     TO ROOM                                    COMH      15
* ----------------------------------------------------                  COMH      16
* HOLE(X,1)    HOLE(X,2)     HOLE(X,3)    HOLE(X,4)                     COMH      17
*   A3            A3            A3            A3                        COMH      18
************************************************************************COMH      19
************************************************************************COMH      20
************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
************************************************************************COMR       3
      INTEGER RMAX                                                      COMR       4
      PARAMETER (RMAX = 20)                                             COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR       6
      INTEGER NROOMS                                                    COMR       7
      REAL ROOM                                                         COMR       8
************************************************************************COMR       9
************************************************************************COMR      10
************************************************************************COMM       1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                     ***COMM       2
************************************************************************COMM       3
      INTEGER MMAX                                                      COMM       4
      PARAMETER (MMAX=100)                                              COMM       5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM       6
     $               MFREQ(MMAX,7), MERR, MTOT                          COMM       7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                              COMM       8
      INTEGER MTOT, MERR                                                COMM       9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                COMM      10
      CHARACTER * 3  MAT                                                COMM      11
      CHARACTER * 70 MATDESC                                            COMM      12
************************************************************************COMM      13
************************************************************************COMM      14
****************************************************************        COMJ       1
*                                                                       COMJ       2
* COMMON FOR EVALUATION OF ROOM MATRIX                                  COMJ       3
*                                                                       COMJ       4
****************************************************************        COMJ       5
      COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME           COMJ       6
     +,SWR(RMAX,6),IDIR                                                 COMJ       7
      REAL TMAT ,ENERGY,POWER,SWR                                       COMJ       8
      LOGICAL FTIME                                                     COMJ       9
************************************************************************COMD       1
```

```
*     COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS         COMD         2
*****************************************************************************COMD     3
      COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ) ,DREFL, DREFLW            COMD         4
      REAL DDABS ,DREFL , DREFLW                                         COMD         5
*****************************************************************************COMD     6
*****************************************************************************COMD     7
*****************************************************                    MASTER      11
*   DECLARATION OF VARIABLES                                             MASTER      12
*****************************************************                    MASTER      13
      INTEGER GETLEN                                                     MASTER      14
*****************************************************                    MASTER      15
*   INITIAL SETUP                                                        MASTER      16
*****************************************************                    MASTER      17
      NROOMS = 3                                                         MASTER      18
      AFLAG = 0                                                          MASTER      19
      RFLAG = 0                                                          MASTER      20
*****************************************************                    MASTER      21
*   INPUT BUILDING IDENTIFICATION                                        MASTER      22
*****************************************************                    MASTER      23
   20 PRINT*, 'ENTER BUILDING IDENTIFICATION  (E.G. ''101'')'            MASTER      24
      PRINT*, '    (NO MORE THAN 5 ALPHANUMERIC CHARACTERS)'             MASTER      25
      READ(1, * , END = 20) BLDG                                         MASTER      26
      PRINT*, 'BUILDING IDENTIFICATION ENTERED AS ''',                   MASTER      27
     $         BLDG(1:GETLEN(BLDG)), ''''                                MASTER      28
 30   PRINT *, 'ENTER NUMBER OF ROOMS IN BUILDING'                       MASTER      29
      READ (1,*,END = 30) NROOMS                                         MASTER      30
*****************************************************                    MASTER      31
*   LOAD ARRAYS FROM DATA FILES                                          MASTER      32
*****************************************************                    MASTER      33
      CALL LMATTER                                                       MASTER      34
      CALL LWALL                                                         MASTER      35
      CALL LTYPE                                                         MASTER      36
      CALL LHOLE                                                         MASTER      37
      CALL LFREQ                                                         MASTER      38
*****************************************************                    MASTER      39
*  CHECK FOR ERROR IN FREQ FILE.                                         MASTER      40
*  IF THERE IS AN ERROR (E.G. MISSING) THEN                              MASTER      41
*  JUST USE THE DEFAULT FREQUENCIES                                      MASTER      42
*****************************************************                    MASTER      43
      IF (FERR.NE.0) THEN                                                MASTER      44
        FTOT = 7                                                         MASTER      45
        DO 40 IFR = 1,7                                                  MASTER      46
   40   FREQA(IFR) = 1.0E03 * (10.0 ** IFR)                              MASTER      47
        PRINT*, ' *** DEFAULT FREQUENCIES WILL BE USED *** '             MASTER      48
      ENDIF                                                              MASTER      49
************************************************************             MASTER      50
      CALL PWALL                                                         MASTER      51
      CALL PHOLE                                                         MASTER      52
      CALL PTYPE                                                         MASTER      53
*****************************************************                    MASTER      54
*   CHECK FOR FILE ERROR                                                 MASTER      55
*****************************************************                    MASTER      56
      IF ( ( MERR .NE. 0 ) .OR. ( WERR .NE. 0 ) .OR.                     MASTER      57
     $     ( HERR .NE. 0 ) .OR. (TERR .NE. 0 ) ) THEN                    MASTER      58
        CALL ERROR (5)                                                   MASTER      59
      END IF                                                             MASTER      60
*****************************************************                    MASTER      61
*                                                                        MASTER      62
*****************************************************                    MASTER      63
      DO 200 IFR= 1,FTOT                                                 MASTER      64
      FREQ = FREQA(IFR)                                                  MASTER      65
      CALL IDDABS                                                        MASTER      66
      CALL LTDB                                                          MASTER      67
      CALL LRAREA                                                        MASTER      68
```

```
      CALL CFACTOR                                   MASTER      69
      CALL DFACTOR                                   MASTER      70
      FTIME = .TRUE.                                 MASTER      71
      DO 100 IDIR = 1 , 5                            MASTER      72
      DO 50 J = 1 , 6                                MASTER      73
   50 POWER (J) = 0.0                                MASTER      74
      POWER (IDIR) = 10.0                            MASTER      75
      CALL ECALC                                     MASTER      76
  100 CALL SPWR                                      MASTER      77
      CALL PPWR2                                     MASTER      78
  200 CONTINUE                                       MASTER      79
      STOP                                           MASTER      80
      END                                            MASTER      81
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
AFLAG	2B	/INITILN/		REAL	
BLDG	0B	/INITILC/		CHAR*5	
DDABS	0B	/ROOMD/		REAL	676
DREFL	1244B	/ROOMD/		REAL	
DREFLW	1245B	/ROOMD/		REAL	
ENERGY	620B	/MAT/		REAL	20
FERR	66B	/INITILN/		INTEGER	
FREQ	0B	/INITILN/		REAL	
FREQA	4B	/INITILN/		REAL	50
FTIME	652B	/MAT/		LOGICAL	
FTOT	67B	/INITILN/		INTEGER	
HERR	1B	/HOLEN/		INTEGER	
HOLE	0B	/HOLEC/		CHAR*3	140
HTOT	0B	/HOLEN/		INTEGER	
IDIR	1043B	/MAT/		INTEGER	
IFR	332B			INTEGER	
J	336B			INTEGER	
MAT	0B	/MATC/		CHAR*3	100
MATDESC	36B	/MATC/		CHAR*70	100
MATTEN	0B	/MATN/		REAL	700
MERR	4374B	/MATN/		INTEGER	
MFREQ	3100B	/MATN/		REAL	700
MRCOEF	1274B	/MATN/		REAL	700
MTOT	4375B	/MATN/		INTEGER	
NROOMS	1244B	/ROOMN/		INTEGER	
POWER	644B	/MAT/		REAL	6
QA	2570B	/MATN/		REAL	100
QR	2734B	/MATN/		REAL	100
QUALITY	1B	/INITILN/		INTEGER	
RAREA	1245B	/ROOMN/		REAL	20
RFLAG	3B	/INITILN/		REAL	
ROOM	0B	/ROOMN/		REAL	676
SWR	653B	/MAT/		REAL	120
TDBTOT	323B	/TYPEN/		INTEGER	
TDB1	37B	/TYPEC/		CHAR*3	35
TDB2	215B	/TYPEN/		REAL	70
TDIM	0B	/TYPEN/		REAL	140
TERR	324B	/TYPEN/		INTEGER	
TMAT	0B	/MAT/		REAL	400
TTOT	214B	/TYPEN/		INTEGER	
TYPE	0B	/TYPEC/		CHAR*3	105
WALL	0B	/WALLC/		CHAR*3	300
WDIM	0B	/WALLN/		REAL	225
WERR	342B	/WALLN/		INTEGER	
WTOT	341B	/WALLN/		INTEGER	

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE      -NAME----TYPE------------------------VALUE

  FMAX    INTEGER                        50        RMAX    INTEGER                        20
  HMAX    INTEGER                        35        TMAX    INTEGER                        35
  MMAX    INTEGER                       100        WMAX    INTEGER                        75


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  CFACTOR                  0      SUBROUTINE       LRAREA                  0      SUBROUTINE
  DFACTOR                  0      SUBROUTINE       LTDB                    0      SUBROUTINE
  ECALC                    0      SUBROUTINE       LTYPE                   0      SUBROUTINE
  ERROR                    1      SUBROUTINE       LWALL                   0      SUBROUTINE
  GETLEN    INTEGER        1      FUNCTION         PHOLE                   0      SUBROUTINE
  IDDABS                   0      SUBROUTINE       PPWR2                   0      SUBROUTINE
  LFREQ                    0      SUBROUTINE       PTYPE                   0      SUBROUTINE
  LHOLE                    0      SUBROUTINE       PWALL                   0      SUBROUTINE
  LMATTER                  0      SUBROUTINE       SPWR                    0      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    20     26B                    148          50  INACTIVE   DO-TERM       198
    30     44B                    153         100  INACTIVE   DO-TERM       201
    40  INACTIVE   DO-TERM        171         200  INACTIVE   DO-TERM       203


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  MASTER     20B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE1   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH          340B =  224
  CM LABELLED COMMON LENGTH  12774B =  5628
  CM STORAGE USED            63000B =  26112
  COMPILE TIME               0.183 SECONDS
```

```
      REAL FUNCTION ATTEN (ID,FREQ,AFLAG)                               ATTEN    1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ATTEN    2
*[[[                                                                 [[[ATTEN    3
*[[[  GIVEN THE FREQUENCY AND THE MATERIAL, THIS FUNCTION RETURNS   [[[ATTEN    4
*[[[  THE ATTENUATION                                                [[[ATTEN    5
*[[[                                                                 [[[ATTEN    6
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[ATTEN    7
***********************************************************************ATTEN    8
*   FREQ IS IN HERTZ                                                    ATTEN    9
*   HIFREQ,LOFREQ, AND F ARE IN LOG ( HERTZ )                           ATTEN   10
*   AFLAG RANGES FROM 0 TO 100 AND DETERMINES HOW MUCH OF THE QUALITY   ATTEN   11
*   FACTOR IS APPLIED TO THE ATTENUATION VALUE.                         ATTEN   12
***********************************************************************ATTEN   13
***********************************************************************COMM     1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                    ***COMM     2
***********************************************************************COMM     3
      INTEGER MMAX                                                      COMM     4
      PARAMETER (MMAX=100)                                              COMM     5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM     6
     $              MFREQ(MMAX,7), MERR, MTOT                           COMM     7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                              COMM     8
      INTEGER MTOT, MERR                                                COMM     9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                COMM    10
      CHARACTER * 3  MAT                                                COMM    11
      CHARACTER * 70 MATDESC                                            COMM    12
***********************************************************************COMM    13
***********************************************************************COMM    14
****************************************************                    ATTEN   15
*   DECLARATION OF VARIABLES                                            ATTEN   16
****************************************************                    ATTEN   17
      REAL FREQ,FRAC,MINFREQ,MAXFREQ,LOFREQ,HIFREQ,LOATTEN,HIATTEN,F    ATTEN   18
      INTEGER R,C,RINDEX,CINDEX                                         ATTEN   19
      LOGICAL FOUND,EXACT                                               ATTEN   20
      CHARACTER *3 ID                                                   ATTEN   21
****************************************************                    ATTEN   22
*   FIND ROW INDEX OF MATERIAL                                          ATTEN   23
****************************************************                    ATTEN   24
      FOUND = .FALSE.                                                   ATTEN   25
      DO 10 R = 1, MMAX                                                 ATTEN   26
         IF ( MAT(R) .EQ. ID ) THEN                                     ATTEN   27
            FOUND = .TRUE.                                              ATTEN   28
            RINDEX = R                                                  ATTEN   29
         END IF                                                         ATTEN   30
10    CONTINUE                                                          ATTEN   31
      IF ( .NOT. FOUND ) THEN                                           ATTEN   32
         IERR = 3                                                       ATTEN   33
         CALL ERROR (IERR)                                              ATTEN   34
      END IF                                                            ATTEN   35
****************************************************                    ATTEN   36
*   TEST FOR FREQUENCY OUT OF RANGE                                     ATTEN   37
****************************************************                    ATTEN   38
      MINFREQ = MFREQ (RINDEX,1)                                        ATTEN   39
      MAXFREQ = MFREQ (RINDEX,7)                                        ATTEN   40
      IF ( FREQ .LT. MINFREQ .OR. FREQ .GT. MAXFREQ ) THEN              ATTEN   41
         IERR = 2                                                       ATTEN   42
         CALL ERROR (IERR)                                              ATTEN   43
      END IF                                                            ATTEN   44
****************************************************                    ATTEN   45
*   IF THE EXACT FREQUENCY IS IN THE TABLE, THEN                        ATTEN   46
*   USE THE ATTENUATION VALUE WITHOUT INTERPOLATION.                    ATTEN   47
****************************************************                    ATTEN   48
      DO 20 C= 1,7                                                      ATTEN   49
      EXACT = .FALSE.                                                   ATTEN   50
      IF ( FREQ .EQ. MFREQ (RINDEX,C) ) THEN                            ATTEN   51
```

```
                EXACT = .TRUE.
                CINDEX = C
                ATTEN = MATTEN (RINDEX,CINDEX)
                ATTEN = ATTEN * ( 1 +  AFLAG / 100 )
             END IF
   20        CONTINUE
*****************************************************
*   INTERPOLATE ATTENUATION VALUES IF EXACT FREQUENCY IS
*   NOT IN THE FREQUENCY/ATTENUATION ARRAYS.
*****************************************************
             IF ( .NOT. EXACT ) THEN
                DO 30 C=1,6
                IF ( FREQ .GT. MFREQ (RINDEX,C) .AND.
     $                FREQ .LT. MFREQ (RINDEX,C+1) )  THEN
                   CINDEX = C
                END IF
   30           CONTINUE
                F = ALOG10 ( FREQ )
                LOFREQ = ALOG10 ( MFREQ (RINDEX, CINDEX) )
                HIFREQ = ALOG10 ( MFREQ (RINDEX, CINDEX + 1) )
                LOATTEN = MATTEN (RINDEX,CINDEX)
                HIATTEN = MATTEN (RINDEX, CINDEX + 1)
                FRAC = (F - LOFREQ) / (HIFREQ - LOFREQ)
                ATTEN = LOATTEN + (FRAC * (HIATTEN - LOATTEN) )
                ATTEN = ATTEN * ( 1 +  AFLAG / 100 )
             END IF
             RETURN
             END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG        3  DUMMY-ARG                  REAL
  ATTEN     242B                             REAL
  C         254B                             INTEGER
  CINDEX    256B                             INTEGER
  EXACT     260B                             LOGICAL
  F         252B                             REAL
  FOUND     257B                             LOGICAL
  FRAC      243B                             REAL
  FREQ         2  DUMMY-ARG                  REAL
  HIATTEN   251B                             REAL
  HIFREQ    247B                             REAL
  ID           1  DUMMY-ARG                  CHAR*3
  IERR      262B                             INTEGER
  LOATTEN   250B                             REAL
  LOFREQ    246B                             REAL
  MAT         0B  /MATC/                     CHAR*3         100
  MATDESC    36B  /MATC/                     CHAR*70        100
  MATTEN      0B  /MATN/                     REAL           700
  MAXFREQ   245B                             REAL
  MERR     4374B  /MATN/                     INTEGER
  MFREQ    3100B  /MATN/                     REAL           700
  MINFREQ   244B                             REAL
  MRCOEF   1274B  /MATN/                     REAL           700
  MTOT     4375B  /MATN/                     INTEGER
  QA       2570B  /MATN/                     REAL           100
  QR       2734B  /MATN/                     REAL           100
  R         253B                             INTEGER
  RINDEX    255B                             INTEGER
```

```
  FUNCTION ATTEN     74/175  OPT=0

--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  MMAX     INTEGER                       100


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ALOG10    REAL            1      INTRINSIC
  ERROR                     1      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        44
     20  INACTIVE   DO-TERM        70
     30  INACTIVE   DO-TERM        81


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ATTEN       6B     3


--STATISTICS--

  PROGRAM-UNIT LENGTH            267B =  183
  CM LABELLED COMMON LENGTH     5730B =  3032
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.132 SECONDS
```

SUBROUTINE ERROR   74/175 OPT=0

```
      SUBROUTINE ERROR(IERR)                                            ERROR        1
      CHARACTER*45 MESSAGE(20)                                          ERROR        2
      DATA MESSAGE( 1)/'MATERIALS DATA BASE IS EMPTY                 '/ ERROR        3
      DATA MESSAGE( 2)/'FREQUENCY IS OUT OF RANGE                    '/ ERROR        4
      DATA MESSAGE( 3)/'THIS MATERIAL IS NOT IN DATA BASE            '/ ERROR        5
      DATA MESSAGE( 4)/'DENOMINATOR IS ZERO                          '/ ERROR        6
      DATA MESSAGE( 5)/'FILE HANDLING ERROR                          '/ ERROR        7
      DATA MESSAGE( 6)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR        8
      DATA MESSAGE( 7)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR        9
      DATA MESSAGE( 8)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       10
      DATA MESSAGE( 9)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       11
      DATA MESSAGE(10)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       12
      DATA MESSAGE(11)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       13
      DATA MESSAGE(12)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       14
      DATA MESSAGE(13)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       15
      DATA MESSAGE(14)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       16
      DATA MESSAGE(15)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       17
      DATA MESSAGE(16)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       18
      DATA MESSAGE(17)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       19
      DATA MESSAGE(18)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       20
      DATA MESSAGE(19)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       21
      DATA MESSAGE(20)/'ERROR CODE IS OUT OF RANGE                   '/ ERROR       22
      IERRM=5                                                           ERROR       23
      IF(IERR.GT.IERRM) IERR=20                                         ERROR       24
      WRITE(6,10) IERR,MESSAGE(IERR)                                    ERROR       25
   10 FORMAT(' ***ERROR NUMBER = ',I5,' ***  ',A45)                     ERROR       26
      CALL PMDSTOP                                                      ERROR       27
      STOP 'ERROR'                                                      ERROR       28
      END                                                               ERROR       29
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  IERR         1   DUMMY-ARG                 INTEGER
  IERRM      210B                            INTEGER
  MESSAGE     56B                            CHAR*45         20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  PMDSTOP                  0      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10     36B     FORMAT          26


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ERROR       5B      1
```

```
 SUBROUTINE ERROR    74/175  OPT=0

--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE6   FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH            213B =  139
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.055 SECONDS
```

```
      SUBROUTINE CFACTOR                                                CFACTOR       1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[CFACTOR       2
*[[[                                                                [[[CFACTOR       3
*[[[ THIS ROUTINE CALCULATES THE ATTENUATION OF EACH WALL AND EACH   [[[CFACTOR       4
*[[[ OPENING IN EACH WALL, LAYER BY LAYER, AND THEN CALCULATES THE   [[[CFACTOR       5
*[[[ COMPOSITE TRANSMISSION FACTORS OF EACH WALL USING AN APPROACH   [[[CFACTOR       6
*[[[ DEVELOPED BY JERRY WYSS.                                        [[[CFACTOR       7
*[[[                                                                [[[CFACTOR       8
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[CFACTOR       9
************************************************************************CFACTOR      10
*** VARIABLE DEFINITIONS:                                               CFACTOR      11
***   WATTEN: WALL ATTENUATION                                          CFACTOR      12
***   OATTEN: OPENING ATTENUATION                                       CFACTOR      13
***   LATTEN: LAYER ATTENUATION                                         CFACTOR      14
***   MATTEN: MATERIAL ATTENUATION                                      CFACTOR      15
***   MAT: MATERIAL IDENTIFICATION                                      CFACTOR      16
***   ID: IDENTIFICATION OF OPENING                                     CFACTOR      17
***   WALL: WALL ARRAY CONTAINING WALL IDENTIFICATION AND MATERIAL      CFACTOR      18
***   WDIM: WALL ARRAY CONTAINING PHYSICAL DIMENSIONS OF THE WALL       CFACTOR      19
***   WMAX: MAXIMUM SIZE OF WALL AND WDIM ARRAYS                        CFACTOR      20
***   WTOT: TOTAL LINES OF DATA IN THE THE WALL AND WDIM ARRAYS.        CFACTOR      21
***   HEIGHT: HEIGHT OF WALL                                            CFACTOR      22
***   WIDTH: WIDTH OF WALL                                              CFACTOR      23
***   WDIM(R,C): THICKNESS OF WALL                                      CFACTOR      24
***   AREA: AREA                                                        CFACTOR      25
***   WAREA: TOTAL WALL AREA WITHOUT SUBTRACTING OPENINGS.              CFACTOR      26
***   OAREA: TOTAL AREA OF THE OPENINGS.                                CFACTOR      27
***   NEWWALL: TRUE IF DATA LINE BELONGS TO A NEW WALL                  CFACTOR      28
***   WALLEND: TRUE IF DATA LINE IS THE LAST DATA LINE OF A WALL        CFACTOR      29
***   T: TRANSMISSION FACTOR                                            CFACTOR      30
***   S: AREA OF OPENING OR WALL AS APPROPRIATE                         CFACTOR      31
***   TS: TRANSMISSION FACTOR = T1*S1 + T2S2 + T3*S3 + ...              CFACTOR      32
***   TS2: TRANSMISSION FACTOR = T1*S1*S1 + T2*S2*S2 + ...              CFACTOR      33
************************************************************************CFACTOR      34
************************************************************************COMF          1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF          2
************************************************************************COMF          3
      INTEGER FMAX                                                      COMF          4
      PARAMETER (FMAX = 50)                                             COMF          5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF          6
     $                 FTOT                                             COMF          7
      COMMON /INITILC/ BLDG                                             COMF          8
      CHARACTER * 5 BLDG                                                COMF          9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF         10
      INTEGER QUALITY, FERR, FTOT                                       COMF         11
************************************************************************COMF         12
************************************************************************COMF         13
************************************************************************COMH          1
***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH          2
************************************************************************COMH          3
      INTEGER HMAX                                                      COMH          4
      PARAMETER (HMAX = 35)                                             COMH          5
      COMMON /HOLEN/ HTOT, HERR                                         COMH          6
      COMMON /HOLEC/ HOLE(HMAX,4)                                       COMH          7
      INTEGER HTOT, HERR                                                COMH          8
      CHARACTER * 3 HOLE                                                COMH          9
* ===================================================                   COMH         10
* DESCRIPTION OF ARRAYS                                                 COMH         11
* ===================================================                   COMH         12
*     ROOM IDENTIFICATION            APERTURE ID                        COMH         13
* -------------------------------------                                 COMH         14
* DIRECTION    FROM ROOM     TO ROOM                                    COMH         15
* ---------------------------------------------------                   COMH         16
* HOLE(X,1)    HOLE(X,2)     HOLE(X,3)    HOLE(X,4)                     COMH         17
```

```
*   A3             A3             A3            A3                           COMH    18
***************************************************************************COMH    19
***************************************************************************COMH    20
***************************************************************************COMT     1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS                 ***COMT     2
***************************************************************************COMT     3
      INTEGER TMAX                                                         COMT     4
      PARAMETER (TMAX=35)                                                  COMT     5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR             COMT     6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                                COMT     7
      INTEGER TTOT,TDBTOT,TERR                                             COMT     8
      REAL TDIM,TDB2                                                       COMT     9
      CHARACTER * 3 TYPE,TDB1                                              COMT    10
*================================================                          COMT    11
* DESCRIPTION OF ARRAYS                                                    COMT    12
*================================================                          COMT    13
*   ID        MATERIAL    FRAME MATERIAL                                   COMT    14
*------------------------------------------------                          COMT    15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                         COMT    16
* A3          A3          A3                                               COMT    17
*================================================                          COMT    18
* HEIGHT      WIDTH       LAYER       DISTANCE                             COMT    19
*                         THICKNESS   ABOVE FLOOR                          COMT    20
*------------------------------------------------                          COMT    21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                            COMT    22
*  F8.2        F8.2        F8.2        F8.2                                COMT    23
*================================================                          COMT    24
*  ID       ATTENUATION    AREA                                            COMT    25
* -----------------------------------------------                          COMT    26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                        COMT    27
*   A3          E9.3         E9.3                                          COMT    28
***************************************************************************COMT    29
***************************************************************************COMT    30
***************************************************************************COMR     1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS             ***COMR     2
***************************************************************************COMR     3
      INTEGER RMAX                                                         COMR     4
      PARAMETER (RMAX = 20)                                                COMR     5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)         COMR     6
      INTEGER NROOMS                                                       COMR     7
      REAL ROOM                                                            COMR     8
***************************************************************************COMR     9
***************************************************************************COMR    10
***************************************************************************COMW     1
***   COMMON FOR DATABASE OF WALL PARAMETERS                            ***COMW     2
***************************************************************************COMW     3
      INTEGER WMAX                                                         COMW     4
      PARAMETER (WMAX = 75)                                                COMW     5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                              COMW     6
      COMMON /WALLC/ WALL(WMAX,4)                                          COMW     7
      INTEGER WTOT,WERR                                                    COMW     8
      REAL WDIM                                                            COMW     9
      CHARACTER *3 WALL                                                    COMW    10
* ================================                                         COMW    11
** DESCRIPTION OF ARRAYS                                                   COMW    12
* ================================                                         COMW    13
*    WALL IDENTIFICATION                                                   COMW    14
* --------------------------------                                         COMW    15
* DIRECTION    FROM        TO                                              COMW    16
*              ROOM        ROOM                                            COMW    17
* --------------------------------                                         COMW    18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                        COMW    19
*  A3          A3          A3                                              COMW    20
* =====================================================                    COMW    21
```

```
*        WALL PARAMETERS                                                 COMW        22
* ----------------------------------------------------------------       COMW        23
* MATERIAL   HEIGHT     WIDTH      LAYER THICKNESS                       COMW        24
* ----------------------------------------------------------------       COMW        25
* WALL(X,4)  WDIM(X,1)  WDIM(X,2)  WDIM(X,3)                             COMW        26
*  A3        F8.2       F8.2       F8.2                                  COMW        27
*************************************************************************COMW        28
*************************************************************************COMW        29
*************************************************************************COMD         1
*     COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS         COMD         2
*************************************************************************COMD         3
      COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ) ,DREFL, DREFLW            COMD         4
      REAL DDABS ,DREFL , DREFLW                                         COMD         5
*************************************************************************COMD         6
*************************************************************************COMD         7
****************************************************                     CFACTOR     41
*   DECLARATION OF VARIABLES                                             CFACTOR     42
****************************************************                     CFACTOR     43
      INTEGER NEXT, LAST, R, C, ROW                                      CFACTOR     44
      REAL WATTEN,LATTEN,OATTEN,MATTEN,ATTEN,T,S,TS,TS2                  CFACTOR     45
      REAL HEIGHT,WIDTH,AREA,OAREA,WAREA                                 CFACTOR     46
      CHARACTER * 3 FROM, TO, MAT, ID                                    CFACTOR     47
      LOGICAL NEWWALL,WALLEND                                            CFACTOR     48
****************************************************                     CFACTOR     49
*   INITIALIZE ROOM MATRIX.                                              CFACTOR     50
****************************************************                     CFACTOR     51
      DO 5 R = 1,RMAX                                                    CFACTOR     52
        DO 5 C = 1,RMAX                                                  CFACTOR     53
          ROOM(R,C) = 0.0                                                CFACTOR     54
    5 CONTINUE                                                           CFACTOR     55
      DO 6 R = 1,RMAX                                                    CFACTOR     56
        DO 6 C = RMAX + 1, RMAX + 5                                      CFACTOR     57
          ROOM(R,C) = 0.0                                                CFACTOR     58
    6 CONTINUE                                                           CFACTOR     59
****************************************************                     CFACTOR     60
*   LOOP & PROCESS EACH LAYER OF THE WALL ARRAY                          CFACTOR     61
****************************************************                     CFACTOR     62
      DO 10 R = 1,WTOT                                                   CFACTOR     63
****************************************************                     CFACTOR     64
*   SET WALLEND CONDITION                                                CFACTOR     65
****************************************************                     CFACTOR     66
        NEXT = R + 1                                                     CFACTOR     67
        IF (R .EQ. WTOT) THEN                                            CFACTOR     68
          WALLEND = .TRUE.                                               CFACTOR     69
        ELSE IF ( WALL(R,2) .NE. WALL(NEXT,2) .OR.                       CFACTOR     70
     $          WALL(R,3) .NE. WALL(NEXT,3) ) THEN                       CFACTOR     71
          WALLEND = .TRUE.                                               CFACTOR     72
        ELSE                                                             CFACTOR     73
          WALLEND = .FALSE.                                              CFACTOR     74
        END IF                                                           CFACTOR     75
****************************************************                     CFACTOR     76
*   SET NEWWALL CONDITION                                                CFACTOR     77
****************************************************                     CFACTOR     78
        LAST = R - 1                                                     CFACTOR     79
        IF (R .EQ. 1) THEN                                               CFACTOR     80
          NEWWALL = .TRUE.                                               CFACTOR     81
        ELSE IF ( WALL(R,2) .NE. WALL(LAST,2) .OR.                       CFACTOR     82
     $          WALL(R,3) .NE. WALL(LAST,3) ) THEN                       CFACTOR     83
          NEWWALL = .TRUE.                                               CFACTOR     84
        ELSE                                                             CFACTOR     85
          NEWWALL = .FALSE.                                              CFACTOR     86
        END IF                                                           CFACTOR     87
****************************************************                     CFACTOR     88
****************************************************                     CFACTOR     89
```

```
*   CALCULATE WALL ATTENUATION, LAYER BY LAYER
**************************************************
         IF (NEWWALL) THEN
**************************************************
*      ...INITIALIZE WALL CONDITIONS
**************************************************
           TS = 0
           TS2 = 0
           WATTEN = 0
         END IF
**************************************************
*      ...CALCULATE ATTENUATION OF LAYER
**************************************************
         MAT = WALL(R,4)
         MATTEN = ATTEN (MAT,FREQ,AFLAG)
         LATTEN = MATTEN * WDIM(R,3)
**************************************************
*      ACCUMULATE ATTENUATION OF WALL FROM LAYERS
**************************************************
         WATTEN = WATTEN + LATTEN
**************************************************
**************************************************
*      CHECK IF END-OF-WALL LAYER
*      AND THEN CALCULATE VALUES FOR HOLES IF TRUE.
*      OTHERWISE GO BACK AND DO THE NEXT LAYER.
**************************************************
         IF (WALLEND) THEN
           FROM = WALL(R,2)
           TO   = WALL(R,3)
**************************************************
*   ...CALCULATE ATTENUATION OF OPENINGS
*   ...AND TOTAL AREA OF OPENINGS
**************************************************
           OAREA = 0
           DO 20 ROW = 1, HTOT
**************************************************
*   ...CHECK FOR A HOLE IN PRESENT WALL
**************************************************
             IF (HOLE(ROW,2) .EQ. FROM .AND. HOLE(ROW,3) .EQ. TO) THEN
**************************************************
*   ...IF THERE IS A MATCH, CALCULATE ITS CONTRIBUTION;
*   ...OTHERWISE KEEP SEARCHING HOLE'S TABLE
**************************************************
               ID = HOLE(ROW,4)
**************************************************
*   ...GET ATTENUATION AND AREA OF HOLE
**************************************************
               CALL SRCHTDB(ID, OATTEN,AREA)
               OAREA = OAREA + AREA
        CALL RESOND (ID)
               IF (OATTEN .LE. 120) THEN
**************************************************
*   ...CALCULATE TRANSMISSION OF HOLE.
*   ...SET TO ZERO IF LESS THAN 120 DB
**************************************************
                 T = 10**((-OATTEN + DREFLW) / 10 )
               ELSE
                 T = 0
               END IF
               S = AREA
**************************************************
*   ...ACCUMULATE TRANSMISSION * AREA AND
*   ...TRANSMISSION * AREA * AREA
*   ...FOR HOLES IN WALL.
```

```
    **************************************************
                   TS = TS + T * S
                   TS2 = TS2 + T * S * S
                 END IF
    20         CONTINUE
    ***************************************************
    **************************************************
    **************************************************
    *   CALCULATE & STORE ATTENUATION OF EACH ROOM
    **************************************************
    *   ...CALCULATE TOTAL WALL AREA
    **************************************************
               HEIGHT = WDIM(R,1)
               WIDTH = WDIM(R,2)
               WAREA = HEIGHT * WIDTH
               S = WAREA - OAREA
    **************************************************
    *   ...CALCULATE ATTENUATION
    **************************************************
               IF (WATTEN .LE. 120.) THEN
    **************************************************
    *       ...CALCULATE TRANSMISSION OF WALL.
    *       ...SET TO ZERO IF LESS THAN -120 DB.
    **************************************************
                 T = 10**(-WATTEN / 10)
               ELSE
                 T = 0
               END IF
               TS = TS + T * S
               TS2 = TS2 + T * S * S
    **************************************************
    *       ...INSERT TOTAL TRANSMISSION * AREA OF WALL INTO ROOM ARRAY
    **************************************************
               CALL LROOM (TS,TS2,FROM,TO)
             END IF
    10     CONTINUE
           RETURN
           END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG       2B  /INITILN/                  REAL
  AREA      500B                             REAL
  BLDG        0B  /INITILC/                  CHAR*5
  C         464B                             INTEGER
  DDABS       0B  /ROOMD/                    REAL          676
  DREFL    1244B  /ROOMD/                    REAL
  DREFLW   1245B  /ROOMD/                    REAL
  FERR       66B  /INITILN/                  INTEGER
  FREQ        0B  /INITILN/                  REAL
  FREQA       4B  /INITILN/                  REAL           50
  FROM      503B                             CHAR*3
  FTOT       67B  /INITILN/                  INTEGER
  HEIGHT    476B                             REAL
  HERR        1B  /HOLEN/                    INTEGER
  HOLE        0B  /HOLEC/                    CHAR*3        140
  HTOT        0B  /HOLEN/                    INTEGER
  ID        506B                             CHAR*3
  LAST      462B                             INTEGER
  LATTEN    467B                             REAL
  MAT       505B                             CHAR*3
  MATTEN    471B                             REAL
```

```
 NEWWALL    507B                               LOGICAL
 NEXT       461B                               INTEGER
 NROOMS    1244B  /ROOMN/                      INTEGER
 OAREA      501B                               REAL
 OATTEN     470B                               REAL
 QUALITY      1B  /INITILN/                    INTEGER
 R          463B                               INTEGER
 RAREA     1245B  /ROOMN/                      REAL          20
 RFLAG        3B  /INITILN/                    REAL
 ROOM         0B  /ROOMN/                      REAL         676
 ROW        465B                               INTEGER
 S          473B                               REAL
 T          472B                               REAL
 TDBTOT     323B  /TYPEN/                      INTEGER
 TDB1        37B  /TYPEC/                      CHAR*3        35
 TDB2       215B  /TYPEN/                      REAL          70
 TDIM         0B  /TYPEN/                      REAL         140
 TERR       324B  /TYPEN/                      INTEGER
 TO         504B                               CHAR*3
 TS         474B                               REAL
 TS2        475B                               REAL
 TTOT       214B  /TYPEN/                      INTEGER
 TYPE         0B  /TYPEC/                      CHAR*3       105
 WALL         0B  /WALLC/                      CHAR*3       300
 WALLEND    510B                               LOGICAL
 WAREA      502B                               REAL
 WATTEN     466B                               REAL
 WDIM         0B  /WALLN/                      REAL         225
 WERR       342B  /WALLN/                      INTEGER
 WIDTH      477B                               REAL
 WTOT       341B  /WALLN/                      INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX     INTEGER                         50
 HMAX     INTEGER                         35
 RMAX     INTEGER                         20
 TMAX     INTEGER                         35
 WMAX     INTEGER                         75


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 ATTEN      REAL            3      FUNCTION
 LROOM                      4      SUBROUTINE
 RESOND                     1      SUBROUTINE
 SRCHTDB                    3      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     5  INACTIVE   DO-TERM       158
     6  INACTIVE   DO-TERM       162
    10  INACTIVE   DO-TERM       292
    20  INACTIVE   DO-TERM       261
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  CFACTOR     5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH              521B =  337
  CM LABELLED COMMON LENGTH       4000B =  2048
  CM STORAGE USED                63000B =  26112
  COMPILE TIME                   0.249 SECONDS
```

```
      SUBROUTINE LHOLE                                                  LHOLE     1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LHOLE     2
*[[[                                                                 [[[LHOLE     3
*[[[   LOAD THE CONTENTS OF THE "HOLE" FILE INTO THE "HOLE" ARRAY    [[[LHOLE     4
*[[[                                                                 [[[LHOLE     5
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LHOLE     6
************************************************************************COMH      1
***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH      2
************************************************************************COMH      3
      INTEGER HMAX                                                      COMH      4
      PARAMETER (HMAX = 35)                                             COMH      5
      COMMON /HOLEN/ HTOT, HERR                                         COMH      6
      COMMON /HOLEC/ HOLE(HMAX,4)                                       COMH      7
      INTEGER HTOT, HERR                                                COMH      8
      CHARACTER * 3 HOLE                                                COMH      9
* ==================================================                    COMH     10
* DESCRIPTION OF ARRAYS                                                 COMH     11
* ==================================================                    COMH     12
*     ROOM IDENTIFICATION               APERTURE ID                     COMH     13
* ----------------------------------                                    COMH     14
* DIRECTION    FROM ROOM     TO ROOM                                    COMH     15
* --------------------------------------------------                    COMH     16
* HOLE(X,1)    HOLE(X,2)     HOLE(X,3)    HOLE(X,4)                     COMH     17
*   A3            A3            A3            A3                        COMH     18
************************************************************************COMH     19
************************************************************************COMH     20
************************************************************************COMF      1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF      2
************************************************************************COMF      3
      INTEGER FMAX                                                      COMF      4
      PARAMETER (FMAX = 50)                                             COMF      5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF      6
     $                 FTOT                                             COMF      7
      COMMON /INITILC/ BLDG                                             COMF      8
      CHARACTER * 5 BLDG                                                COMF      9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF     10
      INTEGER QUALITY, FERR, FTOT                                       COMF     11
************************************************************************COMF     12
************************************************************************COMF     13
      INTEGER GETLEN, R, C                                              LHOLE     9
      CHARACTER * 7  PFN                                                LHOLE    10
      PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'H'                          LHOLE    11
      HERR = 0                                                          LHOLE    12
      CALL PF  ('GET',0,PFN(1:GETLEN(PFN)),'RC',HERR)                   LHOLE    13
      IF ( HERR .EQ. 0 ) THEN                                           LHOLE    14
        OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',                       LHOLE    15
     $         STATUS='OLD', ACCESS='SEQUENTIAL')                       LHOLE    16
1000     FORMAT (1X,4(1X,A3))                                           LHOLE    17
        HTOT = 0                                                        LHOLE    18
        DO 10 R = 1,HMAX                                                LHOLE    19
          READ (3,1000,END=20)(HOLE(R,C),C=1,4)                         LHOLE    20
          HTOT = HTOT + 1                                               LHOLE    21
10      CONTINUE                                                        LHOLE    22
20      CONTINUE                                                        LHOLE    23
        CLOSE(3,STATUS='DELETE')                                        LHOLE    24
      ELSE IF ( HERR .EQ. 2 ) THEN                                      LHOLE    25
        CALL WARNING (1)                                                LHOLE    26
      ELSE                                                              LHOLE    27
        CALL WARNING (2)                                                LHOLE    28
      END IF                                                            LHOLE    29
      RETURN                                                            LHOLE    30
      END                                                               LHOLE    31
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG       2B  /INITILN/                      REAL
  BLDG        0B  /INITILC/                      CHAR*5
  C         214B                                 INTEGER
  FERR       66B  /INITILN/                      INTEGER
  FREQ        0B  /INITILN/                      REAL
  FREQA       4B  /INITILN/                      REAL            50
  FTOT       67B  /INITILN/                      INTEGER
  HERR        1B  /HOLEN/                        INTEGER
  HOLE        0B  /HOLEC/                        CHAR*3         140
  HTOT        0B  /HOLEN/                        INTEGER
  PFN       215B                                 CHAR*7
  QUALITY     1B  /INITILN/                      INTEGER
  R         213B                                 INTEGER
  RFLAG       3B  /INITILN/                      REAL


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  FMAX    INTEGER                         50
  HMAX    INTEGER                         35


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN    INTEGER         1      FUNCTION
  PF                        5      SUBROUTINE
  WARNING                   1      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM       53
     20     73B                   54
   1000    130B     FORMAT        48


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  LHOLE       5B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE3   AUX/FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH             222B =  146
  CM LABELLED COMMON LENGTH       145B =  101
  CM STORAGE USED               61000B =  25088
  COMPILE TIME                  0.094 SECONDS
```

```
      SUBROUTINE LMATTER                                                LMATTER     1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LMATTER     2
*[[[                                                                 [[[LMATTER     3
*[[[   THIS SUBROUTINE LOADS THE MATERIAL DATABASE INTO ARRAYS FOR   [[[LMATTER     4
*[[[   FURTHER PROGRAM USE.                                          [[[LMATTER     5
*[[[                                                                 [[[LMATTER     6
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LMATTER     7
************************************************************************LMATTER     8
************************************************************************COMM        1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                     ***COMM        2
************************************************************************COMM        3
      INTEGER MMAX                                                      COMM        4
      PARAMETER (MMAX=100)                                              COMM        5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM        6
     $              MFREQ(MMAX,7), MERR, MTOT                           COMM        7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                              COMM        8
      INTEGER MTOT, MERR                                                COMM        9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                COMM       10
      CHARACTER * 3  MAT                                                COMM       11
      CHARACTER * 70 MATDESC                                            COMM       12
************************************************************************COMM       13
************************************************************************COMM       14
**************************************************                      LMATTER    10
*   DECLARATION OF VARIABLES                                            LMATTER    11
**************************************************                      LMATTER    12
      INTEGER R, C, VAL                                                 LMATTER    13
      CHARACTER * 3 MATID                                               LMATTER    14
**************************************************                      LMATTER    15
*  GET FILE                                                             LMATTER    16
**************************************************                      LMATTER    17
      MERR = 0                                                          LMATTER    18
      CALL PF ('GET',0,'MATTER','RC',MERR)                              LMATTER    19
**************************************************                      LMATTER    20
*  FILE ERROR CHECK                                                     LMATTER    21
**************************************************                      LMATTER    22
      IF ( MERR .EQ. 0 ) THEN                                           LMATTER    23
999     CONTINUE                                                        LMATTER    24
      ELSE IF ( MERR .EQ. 2 ) THEN                                      LMATTER    25
        CALL WARNING (3)                                                LMATTER    26
        RETURN                                                          LMATTER    27
      ELSE                                                              LMATTER    28
        CALL WARNING (4)                                                LMATTER    29
        RETURN                                                          LMATTER    30
      END IF                                                            LMATTER    31
**************************************************                      LMATTER    32
*   OPEN FILE                                                           LMATTER    33
**************************************************                      LMATTER    34
      OPEN (UNIT = 3, FILE='MATTER',FORM='FORMATTED',                   LMATTER    35
     $      STATUS = 'OLD', ACCESS = 'SEQUENTIAL')                      LMATTER    36
      REWIND (3)                                                        LMATTER    37
**************************************************                      LMATTER    38
*  INITIALIZE ARRAYS                                                    LMATTER    39
**************************************************                      LMATTER    40
      DATA MAT / 100 * '   ' /                                          LMATTER    41
      DATA MATDESC / 100 * '   ' /                                      LMATTER    42
      DATA MFREQ / 700 * 0.0 /                                          LMATTER    43
      DATA MATTEN / 700 * 0.0 /                                         LMATTER    44
      DATA QA / 100 * 0.0 /                                             LMATTER    45
      DATA MRCOEF / 700 * 0.0 /                                         LMATTER    46
      DATA QR / 100 * 0.0 /                                             LMATTER    47
**************************************************                      LMATTER    48
*  READ IN THE MATERIAL FILE                                            LMATTER    49
**************************************************                      LMATTER    50
10    READ (3,1000,END=20) MATID                                        LMATTER    51
```

```
 65          R = VAL(MATID(2:3))                                    LMATTER      52
 66          MAT (R) = MATID                                        LMATTER      53
 67          READ (3,2000,END=20) MATDESC (R)                       LMATTER      54
 68          READ (3,4000,END=20) (MFREQ(R,C),C=1,7)                LMATTER      55
 69          READ (3,4000,END=20) (MATTEN(R,C),C=1,7)               LMATTER      56
 70          READ (3,4000,END=20) QA (R)                            LMATTER      57
 71          READ (3,4000,END=20) (MRCOEF(R,C),C=1,7)               LMATTER      58
 72          READ (3,4000,END=20) QR (R)                            LMATTER      59
 73          GOTO 10                                                LMATTER      60
 74 20       CONTINUE                                               LMATTER      61
 75 1000     FORMAT (A3)                                            LMATTER      62
 76 2000     FORMAT (A70)                                           LMATTER      63
 77 4000     FORMAT (7(1X,E9.3))                                    LMATTER      64
 78 ***************************************************             LMATTER      65
 79 *  CLOSE FILE                                                   LMATTER      66
 80 ***************************************************             LMATTER      67
 81          CLOSE (3,STATUS = 'DELETE')                            LMATTER      68
 82          RETURN                                                 LMATTER      69
 83          END                                                    LMATTER      70
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  C          327B                             INTEGER
  MAT          0B   /MATC/                    CHAR*3        100
  MATDESC     36B   /MATC/                    CHAR*70       100
  MATID      330B                             CHAR*3
  MATTEN       0B   /MATN/                    REAL          700
  MERR      4374B   /MATN/                    INTEGER
  MFREQ     3100B   /MATN/                    REAL          700
  MRCOEF    1274B   /MATN/                    REAL          700
  MTOT      4375B   /MATN/                    INTEGER
  QA        2570B   /MATN/                    REAL          100
  QR        2734B   /MATN/                    REAL          100
  R          326B                             INTEGER

--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  MMAX    INTEGER                      100

--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  PF                      5      SUBROUTINE
  VAL      INTEGER        1      FUNCTION
  WARNING                 1      SUBROUTINE

--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    10     36B                     64         1000    202B     FORMAT          75
    20    160B                     74         2000    204B     FORMAT          76
   999 *NO REFS*                   37         4000    206B     FORMAT          77
```

```
 SUBROUTINE LMATTER  74/175  OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  LMATTER      5B      0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE3   AUX/FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH              336B =  222
  CM LABELLED COMMON LENGTH       5730B =  3032
  CM STORAGE USED                61000B =  25088
  COMPILE TIME                   0.138 SECONDS
```

```
      SUBROUTINE LRAREA                                                 LRAREA    1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LRAREA    2
*[[[                                                                 [[[LRAREA    3
*[[[  CALCULATE THE SURFACE AREA OF EACH ROOM                        [[[LRAREA    4
*[[[ AND INSERT IT IN THE "RAREA" ARRAY                              [[[LRAREA    5
*[[[                                                                 [[[LRAREA    6
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LRAREA    7
************************************************************************COMR    1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR    2
************************************************************************COMR    3
      INTEGER RMAX                                                      COMR    4
      PARAMETER (RMAX = 20)                                             COMR    5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR    6
      INTEGER NROOMS                                                    COMR    7
      REAL ROOM                                                         COMR    8
************************************************************************COMR    9
************************************************************************COMR   10
************************************************************************COMW    1
***   COMMON FOR DATABASE OF WALL PARAMETERS                         ***COMW    2
************************************************************************COMW    3
      INTEGER WMAX                                                      COMW    4
      PARAMETER (WMAX = 75)                                             COMW    5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW    6
      COMMON /WALLC/ WALL(WMAX,4)                                       COMW    7
      INTEGER WTOT,WERR                                                 COMW    8
      REAL WDIM                                                         COMW    9
      CHARACTER *3 WALL                                                 COMW   10
* =================================                                     COMW   11
** DESCRIPTION OF ARRAYS                                                COMW   12
* =================================                                     COMW   13
*    WALL IDENTIFICATION                                                COMW   14
* ---------------------------------                                     COMW   15
* DIRECTION    FROM       TO                                            COMW   16
*              ROOM       ROOM                                          COMW   17
* ---------------------------------                                     COMW   18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                     COMW   19
*  A3          A3          A3                                           COMW   20
* ======================================================                COMW   21
*        WALL PARAMETERS                                                COMW   22
* ------------------------------------------------------                COMW   23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                   COMW   24
* ------------------------------------------------------                COMW   25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                         COMW   26
*  A3          F8.2        F8.2        F8.2                             COMW   27
************************************************************************COMW   28
************************************************************************COMW   29
****************************************************                    LRAREA   10
*   DECLARATION OF VARIABLES                                            LRAREA   11
****************************************************                    LRAREA   12
      INTEGER NEXT, LAST, R, RNUM, VAL                                  LRAREA   13
      CHARACTER * 3 FROM, TO                                            LRAREA   14
      LOGICAL NEWWALL, WALLEND                                          LRAREA   15
****************************************************                    LRAREA   16
*                                                                       LRAREA   17
****************************************************                    LRAREA   18
****************************************************                    LRAREA   19
*   INITIALIZE RAREA MATRIX TO ZERO                                     LRAREA   20
****************************************************                    LRAREA   21
      DO 5 I = 1,RMAX                                                   LRAREA   22
    5 RAREA(I) = 0.0                                                    LRAREA   23
****************************************************                    LRAREA   24
*                                                                       LRAREA   25
****************************************************                    LRAREA   26
      DO 10  R = 1,WTOT                                                 LRAREA   27
```

```
******************************************************
*   SET WALLEND CONDITION
******************************************************
      NEXT = R + 1
      IF (R .EQ. WTOT) THEN
        WALLEND = .TRUE.
      ELSE IF ( WALL(R,2) .NE. WALL(NEXT,2) .OR.
     $          WALL(R,3) .NE. WALL(NEXT,3) ) THEN
        WALLEND = .TRUE.
      ELSE
        WALLEND = .FALSE.
      END IF
******************************************************
*   SET NEWWALL CONDITION
******************************************************
      LAST = R - 1
      IF (R .EQ. 1) THEN
        NEWWALL = .TRUE.
      ELSE IF ( WALL(R,2) .NE. WALL(LAST,2) .OR.
     $          WALL(R,3) .NE. WALL(LAST,3) ) THEN
        NEWWALL = .TRUE.
      ELSE
        NEWWALL = .FALSE.
      END IF
******************************************************
*   INSERT THE AREA INTO THE ARRAY
******************************************************
      IF (NEWWALL) THEN
        FROM = WALL (R,2)
        TO = WALL (R,3)
        IF ( FROM(1:1) .EQ. 'D') THEN
          RNUM = VAL ( TO(1:2) )
          RAREA(RNUM) = RAREA (RNUM) + WDIM(R,1)*WDIM(R,2)
        ELSE IF ( TO(1:1) .EQ. 'D' ) THEN
          RNUM = VAL ( FROM(1:2) )
          RAREA(RNUM) = RAREA (RNUM) + WDIM(R,1)*WDIM(R,2)
        ELSE IF (( FROM(1:1) .NE. 'D' ) .AND. ( TO(1:1) .NE. 'D' )) THEN
          RNUM = VAL (FROM(1:2) )
          RAREA(RNUM) = RAREA (RNUM) + WDIM(R,1)*WDIM(R,2)
          RNUM = VAL ( TO(1:2) )
          RAREA(RNUM) = RAREA (RNUM) + WDIM(R,1)*WDIM(R,2)
        END IF
      END IF
*
10    CONTINUE
      RETURN
      END
```

--VARIABLE MAP--(LO=A)

-NAME-	--ADDRESS--	-BLOCK-	-PROPERTIES-	-TYPE-	-SIZE
FROM	275B			CHAR*3	
I	301B			INTEGER	
LAST	272B			INTEGER	
NEWWALL	277B			LOGICAL	
NEXT	271B			INTEGER	
NROOMS	1244B	/ROOMN/		INTEGER	
R	273B			INTEGER	
RAREA	1245B	/ROOMN/		REAL	20
RNUM	274B			INTEGER	
ROOM	0B	/ROOMN/		REAL	676
TO	276B			CHAR*3	
WALL	0B	/WALLC/		CHAR*3	300

```
 WALLEND   300B                                  LOGICAL
 WDIM        0B  /WALLN/                         REAL          225
 WERR      342B  /WALLN/                         INTEGER
 WTOT      341B  /WALLN/                         INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 RMAX    INTEGER                         20
 WMAX    INTEGER                         75


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 VAL        INTEGER        1      FUNCTION


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     5  INACTIVE   DO-TERM         60
    10  INACTIVE   DO-TERM        109


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 LRAREA      5B     0


--STATISTICS--

 PROGRAM-UNIT LENGTH             306B =  198
 CM LABELLED COMMON LENGTH      1766B =  1014
 CM STORAGE USED               61000B =  25088
 COMPILE TIME                  0.133 SECONDS
```

```
      SUBROUTINE LROOM (TS,TS2,FROM,TO )                               LROOM      1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LROOM      2
*[[[                                                                [[[LROOM      3
*** THIS ROUTINE LOADS THE TRANSMISSION COEFFICIENT INTO THE APPROPRIATELROOM      4
*** LOCATION IN THE 'ROOM' ARRAY.                                      LROOM      5
***                                                                    LROOM      6
*** NROOMS: TOTAL NUMBERS OF ROOMS REPRESENTED BY DATA                 LROOM      7
*** RMAX: MAXIMUM NUMBER POSSIBLE UNDER THE PRESENT PROGRAM CONFIGURATIOLROOM      8
*** TS AND TS2: TRANSMISSION COEFFICIENTS                              LROOM      9
*** FROM: TO: CONTAINS ROOM#'S OR THE DIRECTIONS D1,D2,4,D5,OR D6.     LROOM     10
*[[[                                                                [[[LROOM     11
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LROOM     12
***********************************************************************LROOM     13
***********************************************************************COMR       1
***    COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS         ***COMR       2
***********************************************************************COMR       3
      INTEGER RMAX                                                     COMR       4
      PARAMETER (RMAX = 20)                                            COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)     COMR       6
      INTEGER NROOMS                                                   COMR       7
      REAL ROOM                                                        COMR       8
***********************************************************************COMR       9
***********************************************************************COMR      10
*****************************************************                  LROOM     15
*   DECLARATION OF VARIABLES                                           LROOM     16
*****************************************************                  LROOM     17
      INTEGER VAL, C, R, RNUM, D                                       LROOM     18
      REAL TS, TS2                                                     LROOM     19
      CHARACTER * 3 FROM, TO                                           LROOM     20
*****************************************************                  LROOM     21
*                                                                      LROOM     22
*****************************************************                  LROOM     23
      IF ( FROM(1:1) .EQ. 'D' ) THEN                                   LROOM     24
         RNUM = VAL ( TO(1:2) )                                        LROOM     25
         D = VAL ( FROM(2:2) )                                         LROOM     26
*****************************************************                  LROOM     27
*  INSERT TRANSMISSION COEFFICENT FOR ENERGY ENTERING A ROOM FROM THE  LROOM     28
*  OUTSIDE OF THE BUILDING.                                            LROOM     29
*****************************************************                  LROOM     30
         R = NROOMS + D                                                LROOM     31
         C = RNUM                                                      LROOM     32
         ROOM(R,C) = TS  + ROOM(R,C)                                   LROOM     33
*****************************************************                  LROOM     34
*  INSERT TRANSMISSION COEFFICIENT INTO 'ROOM' ARRAY FOR ENERGY LEAVING LROOM     35
*  A ROOM TO THE OUTSIDE OF THE BUILDING.                              LROOM     36
*****************************************************                  LROOM     37
         R = RNUM                                                      LROOM     38
         C = NROOMS + D                                                LROOM     39
         ROOM(R,C) = TS2 / RAREA(RNUM) + ROOM(R,C)                     LROOM     40
*****************************************************                  LROOM     41
*                                                                      LROOM     42
*****************************************************                  LROOM     43
      ELSE IF ( TO(1:1) .EQ. 'D' ) THEN                                LROOM     44
         RNUM = VAL ( FROM(1:2) )                                      LROOM     45
         D = VAL ( TO(2:2) )                                           LROOM     46
*****************************************************                  LROOM     47
*  INSERT TRANSMISSION COEFFICIENT INTO 'ROOM' ARRAY FOR ENERGY ENTERINGLROOM     48
*  A ROOM FROM THE OUTSIDE OF THE BUILDING.                            LROOM     49
*****************************************************                  LROOM     50
         R = NROOMS + D                                                LROOM     51
         C = RNUM                                                      LROOM     52
         ROOM(R,C) = TS  + ROOM(R,C)                                   LROOM     53
*****************************************************                  LROOM     54
*  INSERT TRANSMISSION COEFFICIENT INTO 'ROOM' ARRAY FOR ENERGY LEAVING LROOM     55
```

```
*  A ROOM TO THE OUTSIDE OF THE BUILDING.                                  LROOM    56
***************************************************                        LROOM    57
          R = RNUM                                                         LROOM    58
          C = NROOMS + D                                                   LROOM    59
          ROOM(R,C) = TS2 / RAREA(RNUM) + ROOM(R,C)                        LROOM    60
***************************************************                        LROOM    61
*                                                                          LROOM    62
***************************************************                        LROOM    63
       ELSE                                                                LROOM    64
***************************************************                        LROOM    65
*  INSERT TRANSMISSION COEFFICIENTS INTO 'ROOM' ARRAY FOR ENERGY GOIN      LROOM    66
*  FROM ROOM TO ROOM.                                                      LROOM    67
***************************************************                        LROOM    68
          R = VAL ( FROM(1:2) )                                            LROOM    69
          C = VAL ( TO(1:2) )                                              LROOM    70
          ROOM(R,C) = TS2 / RAREA(R)  + ROOM(R,C)                          LROOM    71
          ROOM(C,R) = TS2 / RAREA(C)  + ROOM(C,R)                          LROOM    72
       END IF                                                              LROOM    73
       RETURN                                                              LROOM    74
       END                                                                 LROOM    75
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 C          222B                             INTEGER
 D          225B                             INTEGER
 FROM         3  DUMMY-ARG                   CHAR*3
 NROOMS    1244B  /ROOMN/                    INTEGER
 R          223B                             INTEGER
 RAREA     1245B  /ROOMN/                    REAL           20
 RNUM       224B                             INTEGER
 ROOM         0B  /ROOMN/                    REAL          676
 TO           4  DUMMY-ARG                   CHAR*3
 TS           1  DUMMY-ARG                   REAL
 TS2          2  DUMMY-ARG                   REAL

--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE------------------------VALUE

 RMAX    INTEGER                        20

--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 VAL       INTEGER        1      FUNCTION

--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 LROOM      5B     4
```

```
--STATISTICS--

 PROGRAM-UNIT LENGTH              230B =  152
 CM LABELLED COMMON LENGTH       1271B =  697
 CM STORAGE USED                61000B =  25088
 COMPILE TIME                   0.100 SECONDS
```

```
       INTEGER FUNCTION VAL(STRING)                                         LROOM
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LROOM
*[[[                                                                 [[[LROOM
*[[[   RETURNS THE INTEGER VALUE OF A STRING.                        [[[LROOM
*[[[                                                                 [[[LROOM
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LROOM
************************************************************************LROOM
      INTEGER NUMBER,  X, L, EXP, DIGIT, GETLEN                             LROOM
      CHARACTER * (*) STRING                                                LROOM
*                                                                           LROOM
      L = GETLEN(STRING)                                                    LROOM
      NUMBER = 0                                                            LROOM
      DO 10 X = L,1,-1                                                      LROOM
        EXP = L - X                                                         LROOM
        DIGIT = ICHAR(STRING(X:X)) - 16                                     LROOM
        NUMBER = NUMBER + DIGIT*10**EXP                                     LROOM
 10   CONTINUE                                                              LROOM
      VAL = NUMBER                                                          LROOM
      RETURN                                                                LROOM
      END                                                                   LROOM
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  DIGIT     76B                             INTEGER
  EXP       75B                             INTEGER
  L         74B                             INTEGER
  NUMBER    72B                             INTEGER
  STRING     1   DUMMY-ARG                  CHAR*(*)
  VAL       71B                             INTEGER
  X         73B                             INTEGER


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN    INTEGER       1      FUNCTION
  ICHAR     INTEGER       1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        17


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  VAL        6B     1


--STATISTICS--

  PROGRAM-UNIT LENGTH            102B =  66
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.039 SECONDS
```

```
      INTEGER FUNCTION GETLEN (STRING)                                  LROOM         96
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LROOM         97
*[[[                                                                 [[[LROOM         98
*[[[   DETERMINE LENGTH OF STRING EXCLUDING ANY BLANK PADDING        [[[LROOM         99
*[[[                                                                 [[[LROOM        100
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LROOM        101
***********************************************************************LROOM        102
*                                                                       LROOM        103
* ARGUMENT DEFINITIONS --                                               LROOM        104
*    INPUT ARGUMENTS                                                    LROOM        105
*      STRING - STRING WHOSE LENGTH IS TO BE DETERMINED                 LROOM        106
*                                                                       LROOM        107
**************************************************                      LROOM        108
      CHARACTER * (*) STRING                                            LROOM        109
**************************************************                      LROOM        110
*  FUNCTION PARAMETERS                                                  LROOM        111
**************************************************                      LROOM        112
      CHARACTER * 1 BLANK                                               LROOM        113
      PARAMETER (BLANK = ' ')                                           LROOM        114
**************************************************                      LROOM        115
*  LOCAL VARIABLES                                                      LROOM        116
**************************************************                      LROOM        117
      INTEGER NEXT                                                      LROOM        118
**************************************************                      LROOM        119
*  START WITH THE LAST CHARACTER AND FIND THE FIRST NON-BLANK           LROOM        120
**************************************************                      LROOM        121
      DO 10 NEXT = LEN(STRING),1,-1                                     LROOM        122
         IF (STRING(NEXT : NEXT) .NE. BLANK) THEN                       LROOM        123
            GETLEN = NEXT                                               LROOM        124
            RETURN                                                      LROOM        125
         END IF                                                         LROOM        126
   10 CONTINUE                                                          LROOM        127
**************************************************                      LROOM        128
*  ALL CHARACTERS ARE BLANKS                                            LROOM        129
**************************************************                      LROOM        130
      GETLEN = 0                                                        LROOM        131
*                                                                       LROOM        132
      RETURN                                                            LROOM        133
      END                                                               LROOM        134
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  GETLEN     63B                             INTEGER
  NEXT       64B                             INTEGER
  STRING      1   DUMMY-ARG                  CHAR*(*)


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  BLANK   CHAR*1                       ' '


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  LEN       INTEGER        1      INTRINSIC
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM         32


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  GETLEN      6B     1


--STATISTICS--

  PROGRAM-UNIT LENGTH              70B =  56
  CM STORAGE USED               61000B =  25088
  COMPILE TIME                  0.039 SECONDS
```

```
      SUBROUTINE LTDB                                                   LTDB
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LTDB
*[[[                                                                [[[LTDB
*[[[   THIS ROUTINE CALCULATES THE ATTENUATION OF EACH DOOR AND     [[[LTDB
*[[[   WINDOW                                                       [[[LTDB
*[[[                                                                [[[LTDB
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LTDB
***********************************************************************LTDB
* VARIABLE DEFINITIONS                                                  LTDB
*   ID: IDENTIFICATION OF OPENING                                       LTDB
*   OATTEN: OPENING ATTENUATION                                         LTDB
*   LATTEN: LAYER ATTENUATION                                           LTDB
*   MATTEN:  MATERIAL ATTENUATION                                       LTDB
*   MAT: MATERIAL IDENTIFICATION                                        LTDB
*   TDBTOT:  TOTAL LINES OF DATA IN THE TDB1 AND TDB2 ARRAYS            LTDB
*   HEIGHT: HEIGHT OF DOOR OR WINDOW OPENING                            LTDB
*   WIDTH:  WIDTH OF DOOR OR WINDOW                                     LTDB
*   T: THICKNESS OF LAYER                                               LTDB
*   AREA: AREA                                                          LTDB
*   NEWTYPE: TRUE IF DATA LINE BELONGS TO A NEW DOOR OR WINDOW TYPE     LTDB
*   TYPEEND: TRUE IF DATA LINE IS THE LAST DATA LINE OF AN OPENING TYPE LTDB
***********************************************************************LTDB
***********************************************************************COMF
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF
***********************************************************************COMF
      INTEGER FMAX                                                      COMF
      PARAMETER (FMAX = 50)                                             COMF
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF
     $                 FTOT                                             COMF
      COMMON /INITILC/ BLDG                                             COMF
      CHARACTER * 5 BLDG                                                COMF
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF
      INTEGER QUALITY, FERR, FTOT                                       COMF
***********************************************************************COMF
***********************************************************************COMF
***********************************************************************COMT
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT
***********************************************************************COMT
      INTEGER TMAX                                                      COMT
      PARAMETER (TMAX=35)                                               COMT
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT
      INTEGER TTOT,TDBTOT,TERR                                          COMT
      REAL TDIM,TDB2                                                    COMT
      CHARACTER * 3 TYPE,TDB1                                           COMT
*================================================                       COMT
* DESCRIPTION OF ARRAYS                                                 COMT
*================================================                       COMT
*   ID        MATERIAL    FRAME MATERIAL                                COMT
*------------------------------------------------                       COMT
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                      COMT
* A3          A3          A3                                            COMT
*================================================                       COMT
* HEIGHT     WIDTH       LAYER       DISTANCE                           COMT
*                        THICKNESS   ABOVE FLOOR                        COMT
*------------------------------------------------                       COMT
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT
*  F8.2        F8.2        F8.2        F8.2                             COMT
*================================================                       COMT
*  ID      ATTENUATION    AREA                                          COMT
* -----------------------------------------------                       COMT
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                     COMT
*   A3         E9.3         E9.3                                        COMT
***********************************************************************COMT
```

```
*************************************************************************COMT      30
***************************************************                        LTDB      25
*   DECLARATION OF VARIABLES                                               LTDB      26
***************************************************                        LTDB      27
      INTEGER NEXT, LAST, R                                                LTDB      28
      REAL LATTEN, OATTEN, MATTEN, ATTEN                                   LTDB      29
      REAL HEIGHT, WIDTH, AREA                                             LTDB      30
      CHARACTER * 3 MAT , ID                                               LTDB      31
      LOGICAL NEWTYPE, TYPEEND                                             LTDB      32
***************************************************                        LTDB      33
*                                                                          LTDB      34
***************************************************                        LTDB      35
      TDBTOT = 0                                                           LTDB      36
      DO 10 R = 1, TTOT                                                    LTDB      37
***************************************************                        LTDB      38
*   SET TYPEEND CONDITION                                                  LTDB      39
***************************************************                        LTDB      40
      NEXT = R + 1                                                         LTDB      41
      IF (R .EQ. TTOT) THEN                                                LTDB      42
        TYPEEND = .TRUE.                                                   LTDB      43
      ELSE IF ( TYPE(R,1) .NE. TYPE(NEXT,1) ) THEN                         LTDB      44
        TYPEEND = .TRUE.                                                   LTDB      45
      ELSE                                                                 LTDB      46
        TYPEEND = .FALSE.                                                  LTDB      47
      END IF                                                               LTDB      48
***************************************************                        LTDB      49
*   SET NEWTYPE CONDITION                                                  LTDB      50
***************************************************                        LTDB      51
      LAST = R - 1                                                         LTDB      52
      IF (R .EQ. 1) THEN                                                   LTDB      53
        NEWTYPE = .TRUE.                                                   LTDB      54
      ELSE IF ( TYPE(R,1) .NE. TYPE(LAST,1) ) THEN                         LTDB      55
        NEWTYPE = .TRUE.                                                   LTDB      56
      ELSE                                                                 LTDB      57
        NEWTYPE = .FALSE.                                                  LTDB      58
      END IF                                                               LTDB      59
***************************************************                        LTDB      60
*   CALCULATE                                                              LTDB      61
***************************************************                        LTDB      62
      IF (NEWTYPE) THEN                                                    LTDB      63
***************************************************                        LTDB      64
*        ..INITIALIZE TYPE CONDITIONS                                      LTDB      65
***************************************************                        LTDB      66
        OATTEN = 0                                                         LTDB      67
      END IF                                                               LTDB      68
***************************************************                        LTDB      69
*     ...CALCULATE ATTENUATION FACTOR OF LAYER                             LTDB      70
***************************************************                        LTDB      71
      MAT = TYPE(R,2)                                                      LTDB      72
      MATTEN = ATTEN (MAT,FREQ,AFLAG)                                      LTDB      73
      T = TDIM(R,3)                                                        LTDB      74
      LATTEN = MATTEN * T                                                  LTDB      75
***************************************************                        LTDB      76
*     ..CALCULATE RUNNING ATTENUATION FACTOR OF OPENING                    LTDB      77
***************************************************                        LTDB      78
      OATTEN = OATTEN + LATTEN                                             LTDB      79
      IF (TYPEEND) THEN                                                    LTDB      80
***************************************************                        LTDB      81
*      ....CALCULATE TOTAL OPENING AREA                                    LTDB      82
***************************************************                        LTDB      83
         HEIGHT = TDIM(R,1)                                                LTDB      84
         WIDTH = TDIM(R,2)                                                 LTDB      85
         AREA = HEIGHT * WIDTH                                             LTDB      86
***************************************************                        LTDB      87
```

```
*  INSERT ID, ATTENUATION AND AREA INTO TYPE DATABASE ARRAYS       LTDB     88
***************************************************************    LTDB     89
           TDBTOT = TDBTOT + 1                                     LTDB     90
           ID = TYPE(R,1)                                          LTDB     91
           TDB1(TDBTOT) = ID                                       LTDB     92
           TDB2(TDBTOT,1) = OATTEN                                 LTDB     93
           TDB2(TDBTOT,2) = AREA                                   LTDB     94
        END IF                                                     LTDB     95
   10 CONTINUE                                                     LTDB     96
        RETURN                                                     LTDB     97
        END                                                        LTDB     98
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 AFLAG       2B  /INITILN/                  REAL
 AREA      221B                             REAL
 BLDG        0B  /INITILC/                  CHAR*5
 FERR       66B  /INITILN/                  INTEGER
 FREQ        0B  /INITILN/                  REAL
 FREQA       4B  /INITILN/                  REAL             50
 FTOT       67B  /INITILN/                  INTEGER
 HEIGHT    217B                             REAL
 ID        223B                             CHAR*3
 LAST      212B                             INTEGER
 LATTEN    214B                             REAL
 MAT       222B                             CHAR*3
 MATTEN    216B                             REAL
 NEWTYPE   224B                             LOGICAL
 NEXT      211B                             INTEGER
 OATTEN    215B                             REAL
 QUALITY     1B  /INITILN/                  INTEGER
 R         213B                             INTEGER
 RFLAG       3B  /INITILN/                  REAL
 T         227B                             REAL
 TDBTOT    323B  /TYPEN/                    INTEGER
 TDB1       37B  /TYPEC/                    CHAR*3           35
 TDB2      215B  /TYPEN/                    REAL             70
 TDIM        0B  /TYPEN/                    REAL            140
 TERR      324B  /TYPEN/                    INTEGER
 TTOT      214B  /TYPEN/                    INTEGER
 TYPE        0B  /TYPEC/                    CHAR*3          105
 TYPEEND   225B                             LOGICAL
 WIDTH     220B                             REAL

--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE------------------------VALUE

 FMAX    INTEGER                        50
 TMAX    INTEGER                        35

--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 ATTEN     REAL           3      FUNCTION
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM       137


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  LTDB        5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH            232B =  154
  CM LABELLED COMMON LENGTH      470B =  312
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.116 SECONDS
```

```
      SUBROUTINE LTYPE                                                  LTYPE        1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LTYPE        2
*[[[                                                                 [[[LTYPE        3
*[[[  LOAD THE "TYPE" ARRAYS FROM THE TYPE DATA FILE                 [[[LTYPE        4
*[[[                                                                 [[[LTYPE        5
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LTYPE        6
************************************************************************LTYPE        7
*********************************************************************COMT            1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS            ***COMT            2
*********************************************************************COMT            3
      INTEGER TMAX                                                   COMT            4
      PARAMETER (TMAX=35)                                            COMT            5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR       COMT            6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                          COMT            7
      INTEGER TTOT,TDBTOT,TERR                                       COMT            8
      REAL TDIM,TDB2                                                 COMT            9
      CHARACTER * 3 TYPE,TDB1                                        COMT           10
*==================================================                  COMT           11
* DESCRIPTION OF ARRAYS                                              COMT           12
*==================================================                  COMT           13
*   ID        MATERIAL    FRAME MATERIAL                             COMT           14
*--------------------------------------------------                  COMT           15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                  COMT           16
* A3          A3          A3                                         COMT           17
*==================================================                  COMT           18
* HEIGHT      WIDTH       LAYER       DISTANCE                       COMT           19
*                         THICKNESS   ABOVE FLOOR                    COMT           20
*--------------------------------------------------                  COMT           21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                      COMT           22
*  F8.2        F8.2        F8.2        F8.2                          COMT           23
*==================================================                  COMT           24
*  ID       ATTENUATION    AREA                                      COMT           25
* -------------------------------------------------                  COMT           26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                  COMT           27
*   A3          E9.3         E9.3                                    COMT           28
*********************************************************************COMT           29
*********************************************************************COMT           30
*********************************************************************COMF            1
*** COMMON FOR INITIAL PARAMETERS                                  ***COMF            2
*********************************************************************COMF            3
      INTEGER FMAX                                                   COMF            4
      PARAMETER (FMAX = 50)                                          COMF            5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF          6
     $                 FTOT                                          COMF            7
      COMMON /INITILC/ BLDG                                          COMF            8
      CHARACTER * 5 BLDG                                             COMF            9
      REAL FREQ, AFLAG, RFLAG, FREQA                                 COMF           10
      INTEGER QUALITY, FERR, FTOT                                    COMF           11
*********************************************************************COMF           12
*********************************************************************COMF           13
***************************************************                  LTYPE          10
*   DECLARATION OF VARIABLES                                         LTYPE          11
***************************************************                  LTYPE          12
      INTEGER GETLEN, R, C                                           LTYPE          13
      CHARACTER * 7  PFN                                             LTYPE          14
***************************************************                  LTYPE          15
*                                                                    LTYPE          16
***************************************************                  LTYPE          17
      PFN = 'B' // BLDG(1:GETLEN(BLDG)) // 'T'                       LTYPE          18
      TERR = 0                                                       LTYPE          19
      CALL PF ('GET',0,PFN(1ETLEN(PFN)),'RC',TERR)                 LTYPE          20
      IF (TERR .EQ. 0 ) THEN                                         LTYPE          21
        OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',                    LTYPE          22
     $        STATUS='OLD', ACCESS='SEQUENTIAL')                     LTYPE          23
```

```
     65 1000     FORMAT (1X,3(1X,A3),4(1X,F8.2))                               LTYPE      24
     66          TTOT = 0                                                      LTYPE      25
     67          DO 10 R = 1,TMAX                                              LTYPE      26
     68            READ (3,1000,END=20)(TYPE(R,C),C=1,3),(TDIM(R,C),C=1,4)     LTYPE      27
     69            TTOT = TTOT + 1                                             LTYPE      28
     70 10       CONTINUE                                                      LTYPE      29
     71 20       CONTINUE                                                      LTYPE      30
     72          CLOSE(3,STATUS='DELETE')                                      LTYPE      31
     73        ELSE IF ( TERR .EQ. 2 ) THEN                                    LTYPE      32
     74          CALL WARNING (5)                                              LTYPE      33
     75        ELSE                                                            LTYPE      34
     76          CALL WARNING (6)                                              LTYPE      35
     77        END IF                                                          LTYPE      36
     78        RETURN                                                          LTYPE      37
     79        END                                                             LTYPE      38
```

--VARIABLE MAP--(LO=A)

-NAME	---ADDRESS	--BLOCK	-----PROPERTIES	-------TYPE	---------SIZE
AFLAG	2B	/INITILN/		REAL	
BLDG	0B	/INITILC/		CHAR*5	
C	236B			INTEGER	
FERR	66B	/INITILN/		INTEGER	
FREQ	0B	/INITILN/		REAL	
FREQA	4B	/INITILN/		REAL	50
FTOT	67B	/INITILN/		INTEGER	
PFN	237B			CHAR*7	
QUALITY	1B	/INITILN/		INTEGER	
R	235B			INTEGER	
RFLAG	3B	/INITILN/		REAL	
TDBTOT	323B	/TYPEN/		INTEGER	
TDB1	37B	/TYPEC/		CHAR*3	35
TDB2	215B	/TYPEN/		REAL	70
TDIM	0B	/TYPEN/		REAL	140
TERR	324B	/TYPEN/		INTEGER	
TTOT	214B	/TYPEN/		INTEGER	
TYPE	0B	/TYPEC/		CHAR*3	105

--SYMBOLIC CONSTANTS--(LO=A)

-NAME	----TYPE	------------------------VALUE
FMAX	INTEGER	50
TMAX	INTEGER	35

--PROCEDURES--(LO=A)

-NAME	------TYPE	--------ARGS	------CLASS-----
GETLEN	INTEGER	1	FUNCTION
PF		5	SUBROUTINE
WARNING		1	SUBROUTINE

--STATEMENT LABELS--(LO=A)

-LABEL	-ADDRESS	-----PROPERTIES	----DEF
10	INACTIVE	DO-TERM	70
20	111B		71
1000	147B	FORMAT	65

```
SUBROUTINE LTYPE    74/175  OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  LTYPE        5B      0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE3    AUX/FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH             245B =  165
  CM LABELLED COMMON LENGTH       470B =  312
  CM STORAGE USED               63000B =  26112
  COMPILE TIME                  0.103 SECONDS
```

```
      SUBROUTINE LWALL                                                  LWALL      1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LWALL      2
*[[[                                                                [[[LWALL      3
*[[[  LOAD THE CONTENTS OF THE FILE 'WALLS' INTO ARRAYS WALL AND WDIM.  LWALL      4
*[[[                                                                [[[LWALL      5
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LWALL      6
***********************************************************************LWALL      7
**********************************************************************COMW       1
***   COMMON FOR DATABASE OF WALL PARAMETERS                       ***COMW       2
**********************************************************************COMW       3
      INTEGER WMAX                                                    COMW       4
      PARAMETER (WMAX = 75)                                           COMW       5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                         COMW       6
      COMMON /WALLC/ WALL(WMAX,4)                                     COMW       7
      INTEGER WTOT,WERR                                               COMW       8
      REAL WDIM                                                       COMW       9
      CHARACTER *3 WALL                                               COMW      10
* ==================================                                  COMW      11
** DESCRIPTION OF ARRAYS                                              COMW      12
* ==================================                                  COMW      13
*    WALL IDENTIFICATION                                              COMW      14
* ----------------------------------                                  COMW      15
* DIRECTION    FROM       TO                                          COMW      16
*              ROOM       ROOM                                        COMW      17
* ----------------------------------                                  COMW      18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                   COMW      19
*  A3          A3          A3                                         COMW      20
* =========================================================           COMW      21
*        WALL PARAMETERS                                              COMW      22
* ---------------------------------------------------------           COMW      23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                 COMW      24
* ---------------------------------------------------------           COMW      25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                       COMW      26
*  A3         F8.2        F8.2        F8.2                            COMW      27
**********************************************************************COMW      28
**********************************************************************COMW      29
**********************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                  ***COMF       2
**********************************************************************COMF       3
      INTEGER FMAX                                                    COMF       4
      PARAMETER (FMAX = 50)                                           COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR, COMF       6
     $                 FTOT                                           COMF       7
      COMMON /INITILC/ BLDG                                           COMF       8
      CHARACTER * 5 BLDG                                              COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                  COMF      10
      INTEGER QUALITY, FERR, FTOT                                     COMF      11
**********************************************************************COMF      12
**********************************************************************COMF      13
*************************************************                     LWALL     10
*  DECLARATION OF VARIABLES                                           LWALL     11
*************************************************                     LWALL     12
      INTEGER GETLEN, R, C                                            LWALL     13
      CHARACTER * 7 NAME, PFN                                         LWALL     14
*************************************************                     LWALL     15
*                                                                     LWALL     16
*************************************************                     LWALL     17
      NAME = 'B'//BLDG(1:GETLEN(BLDG))//'W'                           LWALL     18
      PFN = NAME (1:GETLEN(NAME))                                     LWALL     19
      WERR = 0                                                        LWALL     20
      CALL PF ('GET',0,PFN(1ETLEN(PFN)),'RC',WERR)                  LWALL     21
      IF ( WERR .EQ. 0 ) THEN                                         LWALL     22
        OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',                     LWALL     23
     $        STATUS='OLD', ACCESS='SEQUENTIAL')                      LWALL     24
```

```
     65 1000       FORMAT (1X,4(1X,A3),3(1X,F8.2))                                  LWALL        25
     66              WTOT = 0                                                       LWALL        26
     67              DO 10 R = 1,WMAX                                               LWALL        27
     68                READ (3,1000,END=20)(WALL(R,C),C=1,4),(WDIM(R,C),C=1,3)      LWALL        28
     69                WTOT = WTOT + 1                                              LWALL        29
     70 10           CONTINUE                                                       LWALL        30
     71 20           CONTINUE                                                       LWALL        31
     72              CLOSE(3,STATUS='DELETE')                                       LWALL        32
     73            ELSE IF ( WERR .EQ. 2 ) THEN                                     LWALL        33
     74              CALL WARNING (7)                                               LWALL        34
     75            ELSE                                                             LWALL        35
     76              CALL WARNING (8)                                               LWALL        36
     77            END IF                                                           LWALL        37
     78            RETURN                                                           LWALL        38
     79            END                                                              LWALL        39


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG       2B   /INITILN/                  REAL
  BLDG        0B   /INITILC/                  CHAR*5
  C         255B                              INTEGER
  FERR       66B   /INITILN/                  INTEGER
  FREQ        0B   /INITILN/                  REAL
  FREQA       4B   /INITILN/                  REAL           50
  FTOT       67B   /INITILN/                  INTEGER
  NAME      256B                              CHAR*7
  PFN       257B                              CHAR*7
  QUALITY     1B   /INITILN/                  INTEGER
  R         254B                              INTEGER
  RFLAG       3B   /INITILN/                  REAL
  WALL        0B   /WALLC/                    CHAR*3        300
  WDIM        0B   /WALLN/                    REAL          225
  WERR      342B   /WALLN/                    INTEGER
  WTOT      341B   /WALLN/                    INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  FMAX     INTEGER                       50
  WMAX     INTEGER                       75


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  GETLEN     INTEGER          1       FUNCTION
  PF                          5       SUBROUTINE
  WARNING                     1       SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM         70
     20    117B                     71
   1000    155B     FORMAT          65
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  LWALL       5B      0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE3   AUX/FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH              265B =  181
  CM LABELLED COMMON LENGTH        566B =  374
  CM STORAGE USED                63000B =  26112
  COMPILE TIME                   0.108 SECONDS
```

```
      SUBROUTINE PHOLE                                                  PHOLE          1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[PHOLE          2
*[[[                                                                 [[[PHOLE          3
*[[[   PRINT OUT THE CONTENTS OF THE HOLE ARRAY                      [[[PHOLE          4
*[[[                                                                 [[[PHOLE          5
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[PHOLE          6
************************************************************************PHOLE          7
************************************************************************COMH           1
***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH           2
************************************************************************COMH           3
      INTEGER HMAX                                                      COMH           4
      PARAMETER (HMAX = 35)                                             COMH           5
      COMMON /HOLEN/ HTOT, HERR                                         COMH           6
      COMMON /HOLEC/ HOLE(HMAX,4)                                       COMH           7
      INTEGER HTOT, HERR                                                COMH           8
      CHARACTER * 3 HOLE                                                COMH           9
* ==================================================                    COMH          10
* DESCRIPTION OF ARRAYS                                                 COMH          11
* ==================================================                    COMH          12
*     ROOM IDENTIFICATION              APERTURE ID                      COMH          13
* -----------------------------------                                   COMH          14
* DIRECTION    FROM ROOM     TO ROOM                                    COMH          15
* --------------------------------------------------                    COMH          16
* HOLE(X,1)    HOLE(X,2)     HOLE(X,3)    HOLE(X,4)                     COMH          17
*   A3           A3            A3           A3                          COMH          18
************************************************************************COMH          19
************************************************************************COMH          20
      INTEGER R,C                                                       PHOLE          9
      PRINT *                                                           PHOLE         10
      PRINT *,' DOOR AND WINDOW LOCATIONS'                              PHOLE         11
      PRINT*,'***************************'                              PHOLE         12
      PRINT*,'       WALL IDENTIFICATION'                               PHOLE         13
      PRINT*,'     ----------------------'                              PHOLE         14
      PRINT*,' ID    DIRECTION  FROM  TO'                               PHOLE         15
      PRINT*,'==== ======================'                              PHOLE         16
      DO 10 R = 1,HTOT                                                  PHOLE         17
        PRINT 1000,HOLE(R,4),(HOLE(R,C),C=1,3)                          PHOLE         18
   10 CONTINUE                                                          PHOLE         19
      PRINT*,'==========================='                              PHOLE         20
 1000 FORMAT (2X,A3,5X,3(3X,A3))                                        PHOLE         21
      RETURN                                                            PHOLE         22
      END                                                               PHOLE         23
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  C          174B                             INTEGER
  HERR         1B  /HOLEN/                    INTEGER
  HOLE         0B  /HOLEC/                    CHAR*3         140
  HTOT         0B  /HOLEN/                    INTEGER
  R          173B                             INTEGER
```

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  HMAX    INTEGER                       35
```

```
--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        38
   1000    124B     FORMAT         40


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  PHOLE       5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH             201B =  129
  CM LABELLED COMMON LENGTH        54B =  44
  CM STORAGE USED               61000B =  25088
  COMPILE TIME                  0.062 SECONDS
```

```
      SUBROUTINE PROOM                                                  PROOM      1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[PROOM      2
*[[[                                                                 [[[PROOM      3
*[[[  PRINTOUT THE CONTENTS OF THE ROOM MATRIX                       [[[PROOM      4
*[[[                                                                 [[[PROOM      5
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[PROOM      6
************************************************************************PROOM      7
************************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF       2
************************************************************************COMF       3
      INTEGER FMAX                                                      COMF       4
      PARAMETER (FMAX = 50)                                             COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF       6
     $                 FTOT                                             COMF       7
      COMMON /INITILC/ BLDG                                             COMF       8
      CHARACTER * 5 BLDG                                                COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF      10
      INTEGER QUALITY, FERR, FTOT                                       COMF      11
************************************************************************COMF      12
************************************************************************COMF      13
************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
************************************************************************COMR       3
      INTEGER RMAX                                                      COMR       4
      PARAMETER (RMAX = 20)                                             COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR       6
      INTEGER NROOMS                                                    COMR       7
      REAL ROOM                                                         COMR       8
************************************************************************COMR       9
************************************************************************COMR      10
*                                                                       PROOM     10
      INTEGER R,C                                                       PROOM     11
*                                                                       PROOM     12
      PRINT*                                                            PROOM     13
      PRINT*,'      ATTENUATION (DB) BETWEEN ROOMS'                     PROOM     14
      PRINT*,'  AT FREQUENCY = ',FREQ,' HERTZ'                          PROOM     15
      PRINT*,'**********************************************************PROOM     16
     $*************'                                                    PROOM     17
      DO 10 R = 1,NROOMS                                                PROOM     18
        PRINT 1000,(ROOM(R,C), C = 1, NROOMS + 6)                       PROOM     19
   10 CONTINUE                                                          PROOM     20
      DO 20 R = NROOMS + 1 , NROOMS  + 6                                PROOM     21
        PRINT 1000, (ROOM(R,C), C = 1, NROOMS )                         PROOM     22
   20 CONTINUE                                                          PROOM     23
      PRINT*,'==========================================================PROOM     24
     $============='                                                    PROOM     25
 1000 FORMAT(1X,12(F8.3) )                                              PROOM     26
      RETURN                                                            PROOM     27
      END                                                               PROOM     28
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG       2B  /INITILN/                  REAL
  BLDG        0B  /INITILC/                  CHAR*5
  C         210B                             INTEGER
  FERR       66B  /INITILN/                  INTEGER
  FREQ        0B  /INITILN/                  REAL
  FREQA       4B  /INITILN/                  REAL           50
  FTOT       67B  /INITILN/                  INTEGER
  NROOMS   1244B  /ROOMN/                    INTEGER
  QUALITY     1B  /INITILN/                  INTEGER
  R         207B                             INTEGER
```

```
 RAREA    1245B  /ROOMN/                      REAL          20
 RFLAG       3B  /INITILN/                    REAL
 ROOM        0B  /ROOMN/                      REAL         676


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX    INTEGER                         50
 RMAX    INTEGER                         20


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        41
    20  INACTIVE   DO-TERM        44
  1000    154B     FORMAT         47


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 PROOM       5B     0


--STATISTICS--

 PROGRAM-UNIT LENGTH              217B =  143
 CM LABELLED COMMON LENGTH       1362B =  754
 CM STORAGE USED                61000B =  25088
 COMPILE TIME                   0.080 SECONDS
```

```
      SUBROUTINE PTDB                                                   PTDB       1
************************************************************************PTDB       2
*  PRINT SUMMARY OF THE ATTENUATION OF THE DOORS AND WINDOWS            PTDB       3
************************************************************************PTDB       4
************************************************************************COMT       1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT       2
************************************************************************COMT       3
      INTEGER TMAX                                                      COMT       4
      PARAMETER (TMAX=35)                                               COMT       5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT       6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT       7
      INTEGER TTOT,TDBTOT,TERR                                          COMT       8
      REAL TDIM,TDB2                                                    COMT       9
      CHARACTER * 3 TYPE,TDB1                                           COMT      10
*=================================================                      COMT      11
* DESCRIPTION OF ARRAYS                                                 COMT      12
*=================================================                      COMT      13
*    ID       MATERIAL    FRAME MATERIAL                                COMT      14
*-------------------------------------------------                      COMT      15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                      COMT      16
* A3          A3          A3                                            COMT      17
*=================================================                      COMT      18
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT      19
*                         THICKNESS   ABOVE FLOOR                       COMT      20
*-------------------------------------------------                      COMT      21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT      22
*  F8.2        F8.2        F8.2        F8.2                             COMT      23
*=================================================                      COMT      24
*  ID       ATTENUATION    AREA                                         COMT      25
* ------------------------------------------------                      COMT      26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                     COMT      27
*   A3          E9.3         E9.3                                       COMT      28
************************************************************************COMT      29
************************************************************************COMT      30
      INTEGER R,C                                                       PTDB       6
      CHARACTER * 3 ID                                                  PTDB       7
      PRINT *                                                           PTDB       8
      PRINT*,'DOOR AND WINDOW SUMMARY'                                  PTDB       9
      PRINT*,'**********************************'                       PTDB      10
      PRINT*,'ID     ATTENUATION     AREA'                              PTDB      11
      PRINT*,'=================================='                       PTDB      12
      DO 10 R = 1 , TDBTOT                                              PTDB      13
      PRINT 1000,TDB1(R),(TDB2(R,C),C=1,2)                              PTDB      14
10    CONTINUE                                                          PTDB      15
      PRINT*,'=================================='                       PTDB      16
1000  FORMAT (1X,A3,5X,F8.2,5X,F6.2)                                    PTDB      17
      RETURN                                                            PTDB      18
      END                                                               PTDB      19
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 C          144B                            INTEGER
 ID         NONE              UNUSED/*S*    CHAR*3
 R          143B                            INTEGER
 TDBTOT     323B  /TYPEN/                   INTEGER
 TDB1        37B  /TYPEC/                   CHAR*3          35
 TDB2       215B  /TYPEN/                   REAL            70
 TDIM         0B  /TYPEN/                   REAL           140
 TERR       324B  /TYPEN/                   INTEGER
 TTOT       214B  /TYPEN/                   INTEGER
 TYPE         0B  /TYPEC/                   CHAR*3         105
```

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  TMAX    INTEGER                        35


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        44
   1000    107B     FORMAT         46


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  PTDB        5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH            151B =  105
  CM LABELLED COMMON LENGTH      377B =  255
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.060 SECONDS
```

```
      SUBROUTINE PTYPE                                                  PTYPE
*CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPTYPE
*CCC                                                                 CCCPTYPE
*CCC   PRINT OUT THE CONTENTS OF THE "TYPE" AND "TDIM" ARRAYS        CCCPTYPE
*CCC                                                                 CCCPTYPE
*CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCPTYPE
************************************************************************PTYPE
************************************************************************COMT
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT
************************************************************************COMT
      INTEGER TMAX                                                      COMT
      PARAMETER (TMAX=35)                                               COMT
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT
      INTEGER TTOT,TDBTOT,TERR                                          COMT
      REAL TDIM,TDB2                                                    COMT
      CHARACTER * 3 TYPE,TDB1                                           COMT
*=================================================                      COMT
* DESCRIPTION OF ARRAYS                                                 COMT
*=================================================                      COMT
*   ID        MATERIAL    FRAME MATERIAL                                COMT
*-------------------------------------------------                      COMT
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                      COMT
* A3          A3          A3                                            COMT
*=================================================                      COMT
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT
*                         THICKNESS   ABOVE FLOOR                       COMT
*-------------------------------------------------                      COMT
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT
*  F8.2        F8.2        F8.2        F8.2                             COMT
*=================================================                      COMT
*  ID       ATTENUATION    AREA                                         COMT
* ------------------------------------------------                      COMT
* TDB1(X)     TDB2(X,1)    TDB2(X,2)                                    COMT
*   A3          E9.3         E9.3                                       COMT
************************************************************************COMT
************************************************************************COMT
*                                                                       PTYPE
      INTEGER  R, C                                                     PTYPE
*                                                                       PTYPE
      PRINT *                                                           PTYPE
      PRINT *,'                 DOOR AND WINDOW PARAMETERS'             PTYPE
      PRINT *,'**** ***********************************************'PTYPE
      PRINT *,' ID   MATERIAL FRAME   HEIGHT WIDTH  LAYER     DISTANCE' PTYPE
      PRINT *,'                MATERIAL               THICKNESS ABOVE FLR'PTYPE
      PRINT *,'==== ==============================================='PTYPE
      DO 10 R = 1,TTOT                                                  PTYPE
        PRINT 1000,(TYPE(R,C),C=1,3),(TDIM(R,C),C=1,4)                  PTYPE
   10 CONTINUE                                                          PTYPE
      PRINT *,'===================================================='PTYPE
 1000 FORMAT (2X,A3,4X,A3,6X,A3,4(1X,F7.2))                             PTYPE
      RETURN                                                            PTYPE
      END                                                               PTYPE
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
C	217B			INTEGER	
R	216B			INTEGER	
TDBTOT	323B	/TYPEN/		INTEGER	
TDB1	37B	/TYPEC/		CHAR*3	35
TDB2	215B	/TYPEN/		REAL	70
TDIM	0B	/TYPEN/		REAL	140

```
 TERR       324B  /TYPEN/                            INTEGER
 TTOT       214B  /TYPEN/                            INTEGER
 TYPE         0B  /TYPEC/                            CHAR*3         105


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-------------------------VALUE

  TMAX     INTEGER                          35


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM         49
   1000     152B    FORMAT          51


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  PTYPE        5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH             225B =  149
  CM LABELLED COMMON LENGTH       377B =  255
  CM STORAGE USED               61000B =  25088
  COMPILE TIME                  0.069 SECONDS
```

```
      SUBROUTINE PWALL                                                  PWALL       1
**************************************************                     PWALL       2
*** PRINT OUT THE CONTENTS OF THE WALL AND WDIM ARRAYS                  PWALL       3
**************************************************                     PWALL       4
*                                                                       PWALL       5
************************************************************************COMW        1
***    COMMON FOR DATABASE OF WALL PARAMETERS                        ***COMW        2
************************************************************************COMW        3
      INTEGER WMAX                                                      COMW        4
      PARAMETER (WMAX = 75)                                             COMW        5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW        6
      COMMON /WALLC/ WALL(WMAX,4)                                       COMW        7
      INTEGER WTOT,WERR                                                 COMW        8
      REAL WDIM                                                         COMW        9
      CHARACTER *3 WALL                                                 COMW       10
* =================================                                     COMW       11
** DESCRIPTION OF ARRAYS                                                COMW       12
* =================================                                     COMW       13
*    WALL IDENTIFICATION                                                COMW       14
* ---------------------------------                                     COMW       15
* DIRECTION    FROM       TO                                            COMW       16
*              ROOM       ROOM                                          COMW       17
* ---------------------------------                                     COMW       18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                     COMW       19
*  A3          A3          A3                                           COMW       20
* =========================================================             COMW       21
*        WALL PARAMETERS                                                COMW       22
* ---------------------------------------------------------             COMW       23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                   COMW       24
* ---------------------------------------------------------             COMW       25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                         COMW       26
*  A3         F8.2        F8.2        F8.2                              COMW       27
************************************************************************COMW       28
************************************************************************COMW       29
*                                                                       PWALL       7
      INTEGER R, C                                                      PWALL       8
*                                                                       PWALL       9
      PRINT*                                                            PWALL      10
      PRINT *,'WALL IDENTIFICATION              WALL PARAMETERS'        PWALL      11
      PRINT *,'******************** ************************************'PWALL      12
      PRINT *,' DIR  FROM TO         MATERIAL HEIGHT  WIDTH   THICKNESS'PWALL      13
      PRINT *,'=================== ===================================='PWALL      14
      DO 10 R = 1,WTOT                                                  PWALL      15
        PRINT 1000,(WALL(R,C),C=1,4),(WDIM(R,C),C=1,3)                  PWALL      16
   10 CONTINUE                                                          PWALL      17
      PRINT *,'========================================================'PWALL      18
 1000 FORMAT (1X,3(2X,A3),10X,A3,1X,3(1X,F7.2))                         PWALL      19
      RETURN                                                            PWALL      20
      END                                                               PWALL      21
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  C          205B                            INTEGER
  R          204B                            INTEGER
  WALL         0B  /WALLC/                   CHAR*3         300
  WDIM         0B  /WALLN/                   REAL           225
  WERR       342B  /WALLN/                   INTEGER
  WTOT       341B  /WALLN/                   INTEGER
```

```
--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  WMAX     INTEGER                        75


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10   INACTIVE   DO-TERM        45
   1000     142B     FORMAT         47


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  PWALL       5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH            213B =  139
  CM LABELLED COMMON LENGTH      475B =  317
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.063 SECONDS
```

```
      REAL FUNCTION RCOEF (MATID,FREQ,RFLAG)                            RCOEF       1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[RCOEF       2
*[[[                                                                 [[[RCOEF       3
*[[[   GIVEN THE MATERIAL AND THE FREQUENCY, THIS FUNCTION RETURNS   [[[RCOEF       4
*[[[   THE REFLECTION COEFFICIENT                                    [[[RCOEF       5
*[[[                                                                 [[[RCOEF       6
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[RCOEF       7
************************************************************************RCOEF       8
*   FREQ IS IN HERTZ                                                    RCOEF       9
*   HIFREQ,LOFREQ, AND F ARE IN LOG ( HERTZ )                           RCOEF      10
*   RFLAG RANGES FROM 0 TO 100 AND DETERMINES HOW MUCH OF THE QUALITY   RCOEF      11
*   FACTOR IS APPLIED TO THE REFLECTION COEFFICIENT VALUE.              RCOEF      12
************************************************************************RCOEF      13
************************************************************************COMM        1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                     ***COMM        2
************************************************************************COMM        3
      INTEGER MMAX                                                      COMM        4
      PARAMETER (MMAX=100)                                              COMM        5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM        6
     $              MFREQ(MMAX,7), MERR, MTOT                           COMM        7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                              COMM        8
      INTEGER MTOT, MERR                                                COMM        9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                COMM       10
      CHARACTER * 3  MAT                                                COMM       11
      CHARACTER * 70 MATDESC                                            COMM       12
************************************************************************COMM       13
************************************************************************COMM       14
*************************************************                       RCOEF      15
*   DECLARATION OF VARIABLES                                            RCOEF      16
*************************************************                       RCOEF      17
      REAL FREQ,FRAC,MINFREQ,MAXFREQ,LOFREQ,HIFREQ,LORCOEF,HIRCOEF,F    RCOEF      18
      INTEGER R,C,RINDEX,CINDEX                                         RCOEF      19
      LOGICAL FOUND,EXACT                                               RCOEF      20
      CHARACTER *3 MATID                                                RCOEF      21
*************************************************                       RCOEF      22
*   FIND ROW INDEX OF MATERIAL                                          RCOEF      23
*************************************************                       RCOEF      24
      FOUND = .FALSE.                                                   RCOEF      25
      DO 10 R = 1, MMAX                                                 RCOEF      26
         IF ( MAT(R) .EQ. MATID ) THEN                                  RCOEF      27
            FOUND = .TRUE.                                              RCOEF      28
            RINDEX = R                                                  RCOEF      29
         END IF                                                         RCOEF      30
10    CONTINUE                                                          RCOEF      31
      IF ( .NOT. FOUND ) THEN                                           RCOEF      32
         IERR = 3                                                       RCOEF      33
         CALL ERROR (IERR)                                              RCOEF      34
      END IF                                                            RCOEF      35
*************************************************                       RCOEF      36
*   TEST FOR FREQUENCY OUT OF RANGE                                     RCOEF      37
*************************************************                       RCOEF      38
      MINFREQ = MFREQ (RINDEX,1)                                        RCOEF      39
      MAXFREQ = MFREQ (RINDEX,7)                                        RCOEF      40
      IF ( FREQ .LT. MINFREQ .OR. FREQ .GT. MAXFREQ ) THEN              RCOEF      41
         IERR = 2                                                       RCOEF      42
         CALL ERROR (IERR)                                              RCOEF      43
      END IF                                                            RCOEF      44
*************************************************                       RCOEF      45
*   IF THE EXACT FREQUENCY IS IN THE TABLE, THEN                        RCOEF      46
*   USE THE REFLECTION COEFFICIENT VALUE WITHOUT INTERPOLATION.         RCOEF      47
*************************************************                       RCOEF      48
      DO 20 C= 1,7                                                      RCOEF      49
         EXACT = .FALSE.                                                RCOEF      50
         IF ( FREQ .EQ. MFREQ (RINDEX,C) ) THEN                         RCOEF      51
```

```
                    EXACT = .TRUE.                                             RCOEF
                    CINDEX = C                                                 RCOEF
                    RCOEF = MRCOEF (RINDEX,CINDEX)                             RCOEF
                    RCOEF = RCOEF * ( 1 +  RFLAG / 100 )                       RCOEF
                 END IF                                                        RCOEF
   20     CONTINUE                                                             RCOEF
*****************************************************                          RCOEF
*    INTERPOLATE REFLECTION COEFFICIENT VALUES IF EXACT FREQUENCY IS           RCOEF
*    NOT IN THE FREQUENCY/REFLECTION COEFFICIENT ARRAYS.                       RCOEF
*****************************************************                          RCOEF
          IF ( .NOT. EXACT ) THEN                                              RCOEF
             DO 30 C=1,6                                                       RCOEF
                IF ( FREQ .GT. MFREQ (RINDEX,C) .AND.                          RCOEF
     $               FREQ .LT. MFREQ (RINDEX,C+1) )  THEN                      RCOEF
                   CINDEX = C                                                  RCOEF
                END IF                                                         RCOEF
   30        CONTINUE                                                          RCOEF
             F = ALOG10 ( FREQ )                                               RCOEF
             LOFREQ = ALOG10 ( MFREQ (RINDEX, CINDEX) )                        RCOEF
             HIFREQ = ALOG10 ( MFREQ (RINDEX, CINDEX + 1) )                    RCOEF
             LORCOEF = MRCOEF (RINDEX,CINDEX)                                  RCOEF
             HIRCOEF = MRCOEF (RINDEX, CINDEX + 1)                             RCOEF
             FRAC = (F - LOFREQ) / (HIFREQ - LOFREQ)                           RCOEF
             RCOEF = LORCOEF + (FRAC * (HIRCOEF - LORCOEF) )                   RCOEF
             RCOEF = RCOEF * ( 1 +  RFLAG / 100 )                              RCOEF
          END IF                                                               RCOEF
          RETURN                                                               RCOEF
          END                                                                  RCOEF
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
C	254B			INTEGER	
CINDEX	256B			INTEGER	
EXACT	260B			LOGICAL	
F	252B			REAL	
FOUND	257B			LOGICAL	
FRAC	243B			REAL	
FREQ	2	DUMMY-ARG		REAL	
HIFREQ	247B			REAL	
HIRCOEF	251B			REAL	
IERR	262B			INTEGER	
LOFREQ	246B			REAL	
LORCOEF	250B			REAL	
MAT	0B	/MATC/		CHAR*3	100
MATDESC	36B	/MATC/		CHAR*70	100
MATID	1	DUMMY-ARG		CHAR*3	
MATTEN	0B	/MATN/		REAL	700
MAXFREQ	245B			REAL	
MERR	4374B	/MATN/		INTEGER	
MFREQ	3100B	/MATN/		REAL	700
MINFREQ	244B			REAL	
MRCOEF	1274B	/MATN/		REAL	700
MTOT	4375B	/MATN/		INTEGER	
QA	2570B	/MATN/		REAL	100
QR	2734B	/MATN/		REAL	100
R	253B			INTEGER	
RCOEF	242B			REAL	
RFLAG	3	DUMMY-ARG		REAL	
RINDEX	255B			INTEGER	

```
--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE------------------------VALUE

 MMAX     INTEGER                        100


--PROCEDURES--(LO=A)
-NAME------TYPE--------ARGS------CLASS-----

 ALOG10    REAL            1      INTRINSIC
 ERROR                     1      SUBROUTINE


--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        44
    20  INACTIVE   DO-TERM        70
    30  INACTIVE   DO-TERM        81


--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 RCOEF       6B     3


--STATISTICS--

 PROGRAM-UNIT LENGTH           267B =  183
 CM LABELLED COMMON LENGTH    5730B =  3032
 CM STORAGE USED             61000B =  25088
 COMPILE TIME                0.127 SECONDS
```

```
      SUBROUTINE SRCHTDB (ID, OATTEN,OAREA)                             SRCHTDB    1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[SRCHTDB    2
*[[[                                                                 [[[SRCHTDB    3
*[[[   GIVEN AN OPENING ID, THIS SUBROUTINE RETURNS ITS ATTENUATION  [[[SRCHTDB    4
*[[[   AND AREA.                                                     [[[SRCHTDB    5
*[[[                                                                 [[[SRCHTDB    6
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[SRCHTDB    7
************************************************************************COMT       1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT       2
************************************************************************COMT       3
      INTEGER TMAX                                                      COMT       4
      PARAMETER (TMAX=35)                                               COMT       5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT       6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT       7
      INTEGER TTOT,TDBTOT,TERR                                          COMT       8
      REAL TDIM,TDB2                                                    COMT       9
      CHARACTER * 3 TYPE,TDB1                                           COMT      10
*=================================================                      COMT      11
* DESCRIPTION OF ARRAYS                                                 COMT      12
*=================================================                      COMT      13
*   ID        MATERIAL    FRAME MATERIAL                                COMT      14
*-------------------------------------------------                      COMT      15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                      COMT      16
* A3          A3          A3                                            COMT      17
*=================================================                      COMT      18
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT      19
*                         THICKNESS   ABOVE FLOOR                       COMT      20
*-------------------------------------------------                      COMT      21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT      22
*  F8.2        F8.2        F8.2        F8.2                             COMT      23
*=================================================                      COMT      24
*  ID       ATTENUATION    AREA                                         COMT      25
* ------------------------------------------------                      COMT      26
* TDB1(X)    TDB2(X,1)    TDB2(X,2)                                     COMT      27
*   A3          E9.3         E9.3                                       COMT      28
************************************************************************COMT      29
************************************************************************COMT      30
      INTEGER R                                                         SRCHTDB    9
      REAL OAREA,OATTEN                                                 SRCHTDB   10
      CHARACTER *3 ID                                                   SRCHTDB   11
*                                                                       SRCHTDB   12
      DO 10 R = 1 , TDBTOT                                              SRCHTDB   13
      IF (TDB1(R) .EQ. ID) THEN                                         SRCHTDB   14
         OATTEN = TDB2(R,1)                                             SRCHTDB   15
         OAREA = TDB2(R,2)                                              SRCHTDB   16
         RETURN                                                         SRCHTDB   17
      END IF                                                            SRCHTDB   18
   10 CONTINUE                                                          SRCHTDB   19
      END                                                               SRCHTDB   20
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ID           1   DUMMY-ARG                  CHAR*3
  OAREA        3   DUMMY-ARG                  REAL
  OATTEN       2   DUMMY-ARG                  REAL
  R          65B                              INTEGER
  TDBTOT    323B  /TYPEN/                     INTEGER
  TDB1       37B  /TYPEC/                     CHAR*3          35
  TDB2      215B  /TYPEN/                     REAL            70
  TDIM        0B  /TYPEN/                     REAL           140
  TERR      324B  /TYPEN/                     INTEGER
  TTOT      214B  /TYPEN/                     INTEGER
```

```
SUBROUTINE SRCHTDB  74/175  OPT=0

 TYPE         0B  /TYPEC/                     CHAR*3        105


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE-------------------------VALUE

 TMAX     INTEGER                         35


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

     10  INACTIVE   DO-TERM        48


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 SRCHTDB      5B     3


--STATISTICS--

 PROGRAM-UNIT LENGTH             71B =  57
 CM LABELLED COMMON LENGTH      377B =  255
 CM STORAGE USED              61000B =  25088
 COMPILE TIME                 0.049 SECONDS
```

SUBROUTINE WARNING 74/175 OPT=0

```
      1        SUBROUTINE WARNING(ERR)                                          WARNING        1
      2        INTEGER ERR, ERRM                                                WARNING        2
      3        CHARACTER*45 MESSAGE(20)                                         WARNING        3
      4        DATA MESSAGE( 1)/'"HOLE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING       4
      5        DATA MESSAGE( 2)/'FILE HANDLING PROBLEM ON "HOLE" DATA FILE    '/ WARNING       5
      6        DATA MESSAGE( 3)/'"MATTER" FILE DOES NOT EXIST FOR THIS BLDG   '/ WARNING       6
      7        DATA MESSAGE( 4)/'FILE HANDLING PROBLEM ON "MATTER FILE        '/ WARNING       7
      8        DATA MESSAGE( 5)/'"TYPE" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING       8
      9        DATA MESSAGE( 6)/'FILE HANDLING PROBLEM ON "TYPE" FILE         '/ WARNING       9
     10        DATA MESSAGE( 7)/'"WALL" DATA FILE DOES NOT EXIST FOR THIS BLDG'/ WARNING      10
     11        DATA MESSAGE( 8)/'FILE HANDLING PROBLEM ON "WALL" FILE         '/ WARNING      11
     12        DATA MESSAGE( 9)/'HEIGHT AND WIDTH OF ROOM MISSING             '/ WARNING      12
     13        DATA MESSAGE(10)/'LENGTH OF ROOM IS MISSING                    '/ WARNING      13
     14        DATA MESSAGE(11)/'FREQ FILE DOES NOT EXIST FOR THIS BLDG       '/ WARNING      14
     15        DATA MESSAGE(12)/'FILE HANDLING PROBLEM WITH FREQ FILE         '/ WARNING      15
     16        DATA MESSAGE(13)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING      16
     17        DATA MESSAGE(14)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING      17
     18        DATA MESSAGE(15)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING      18
     19        DATA MESSAGE(16)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING      19
     20        DATA MESSAGE(17)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING      20
     21        DATA MESSAGE(18)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING      21
     22        DATA MESSAGE(19)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING      22
     23        DATA MESSAGE(20)/'WARNING CODE IS OUT OF RANGE                 '/ WARNING      23
     24        ERRM=12                                                          WARNING       24
     25        IERR = ERR                                                       WARNING       25
     26        IF(ERR.GT.ERRM) IERR=20                                          WARNING       26
     27        WRITE(6,20)                                                      WARNING       27
     28        WRITE(6,10) ERR,MESSAGE(IERR)                                    WARNING       28
     29        WRITE(6,20)                                                      WARNING       29
     30 10     FORMAT(' ***WARNING NUMBER = ',I5,' ***  ',A45)                  WARNING       30
     31 20     FORMAT('     ')                                                  WARNING       31
     32        RETURN                                                           WARNING       32
     33        END                                                              WARNING       33
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  ERR          1   DUMMY-ARG                 INTEGER
  ERRM       60B                             INTEGER
  IERR      213B                             INTEGER
  MESSAGE    61B                             CHAR*45         20


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10      34B     FORMAT          30
    20      42B     FORMAT          31


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  WARNING     5B     1
```

```
--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

  TAPE6    FMT/SEQ


--STATISTICS--

  PROGRAM-UNIT LENGTH              216B =  142
  CM STORAGE USED                61000B =  25088
  COMPILE TIME                   0.061 SECONDS
```

```
      SUBROUTINE SETUP                                                 SETUP      1
***********************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS         ***COMR       2
***********************************************************************COMR       3
      INTEGER RMAX                                                     COMR       4
      PARAMETER (RMAX = 20)                                            COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)     COMR       6
      INTEGER NROOMS                                                   COMR       7
      REAL ROOM                                                        COMR       8
***********************************************************************COMR       9
***********************************************************************COMR      10
*****************************************************************      COMJ       1
*                                                                      COMJ       2
* COMMON FOR EVALUATION OF ROOM MATRIX                                 COMJ       3
*                                                                      COMJ       4
*****************************************************************      COMJ       5
      COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME          COMJ       6
     +,SWR(RMAX,6),IDIR                                                COMJ       7
      REAL TMAT ,ENERGY,POWER,SWR                                      COMJ       8
      LOGICAL FTIME                                                    COMJ       9
***********************************************************************COMD       1
*     COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS       COMD       2
***********************************************************************COMD       3
      COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ) ,DREFL, DREFLW          COMD       4
      REAL DDABS ,DREFL , DREFLW                                       COMD       5
***********************************************************************COMD       6
***********************************************************************COMD       7
      DATA FTIME /.TRUE./                                              SETUP      5
      IF (FTIME.EQV..FALSE.) GOTO 500                                  SETUP      6
C****************************************************                  SETUP      7
C* CALCULATE DIAGONAL ELEMENTS                                         SETUP      8
C* ASSUME DIAGONAL ELEMENTS ARE INITIALLY ZERO                         SETUP      9
C****************************************************                  SETUP     10
         DO 200 IR=1,NROOMS                                            SETUP     11
         DIAG =0.0                                                     SETUP     12
              DO 100 IC=1,NROOMS + 6                                   SETUP     13
  100         DIAG =DIAG + ROOM(IR,IC) + DDABS(IR,IC)                  SETUP     14
  200    ROOM(IR,IR) = -DIAG                                           SETUP     15
C***************************************************                   SETUP     16
C* SET FTIME FALSE                                                     SETUP     17
C***************************************************                   SETUP     18
      FTIME = .FALSE.                                                  SETUP     19
C***************************************************                   SETUP     20
                                                                       SETUP     21
C* NOW LOAD ROOM INTO TMAT                                             SETUP     22
C* NOTE THAT THE T MATRIX IS REFLECTED ABOUT THE DIAGONAL              SETUP     23
C*  WITH RESPECT TO THE ROOM MATRIX                                    SETUP     24
C***************************************************                   SETUP     25
  500    DO 600 IR = 1, NROOMS                                         SETUP     26
         DO 600 IC = 1, NROOMS                                         SETUP     27
  600    TMAT(IR,IC) = ROOM(IC,IR)                                     SETUP     28
      RETURN                                                           SETUP     29
      END                                                              SETUP     30
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  DDABS       0B  /ROOMD/                    REAL          676
  DIAG      114B                             REAL
  DREFL    1244B  /ROOMD/                    REAL
  DREFLW   1245B  /ROOMD/                    REAL
  ENERGY    620B  /MAT/                      REAL           20
  FTIME     652B  /MAT/                      LOGICAL
```

```
 IC          115B                              INTEGER
 IDIR       1043B  /MAT/                       INTEGER
 IR          112B                              INTEGER
 NROOMS     1244B  /ROOMN/                     INTEGER
 POWER       644B  /MAT/                       REAL            6
 RAREA      1245B  /ROOMN/                     REAL           20
 ROOM          0B  /ROOMN/                     REAL          676
 SWR         653B  /MAT/                       REAL          120
 TMAT          0B  /MAT/                       REAL          400


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 RMAX     INTEGER                         20


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    100  INACTIVE   DO-TERM         37
    200  INACTIVE   DO-TERM         38
    500     53B                     49
    600  INACTIVE   DO-TERM         51


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 SETUP        5B     0


--STATISTICS--

 PROGRAM-UNIT LENGTH              123B =  83
 CM LABELLED COMMON LENGTH       3603B =  1923
 CM STORAGE USED                61000B =  25088
 COMPILE TIME                   0.074 SECONDS
```

```
C**       FUNCTION DETERM
C**
C**       PURPOSE
C**         CALCULATE THE DETERMINANT OF A SQUARE MATRIX
C**
C**       USAGE
C**         DET=DETERM(ARRAY,NORDER)
C**
C**       DESCRIPTION OF PARAMETERS
C**         ARRAY  - MATRIX
C**         NORDER - ORDER OF DETERMINANT(DEGREE OF MATRIX)
C**
C**       COMMENTS
C**         THIS SUBROUTINE DESTROYS THE INPUT MATRIX ARRAY
C**         THIS ROUTINE WAS MODIFIED SO THAT THE MAXIMUM
C**         VALUE IN THE TOP ROW IS MOVED OVER TO THE DIAGONAL
      FUNCTION DETERM(ARRAY,NORDER)
      DIMENSION ARRAY(20,*)
   10 DETERM=1.
   11 DO 50 K=1,NORDER
C**
C**       INTERCHANGE COLUMNS IF DIAGONAL ELEMENT IS ZERO
C**
   21 AMAX=0.0
      JMAX=K
      DO 25 J=K,NORDER
      TMP=ARRAY(K,J)
      TMP=ABS(TMP)
      IF(TMP.LT.AMAX) GOTO 25
      AMAX=TMP
      JMAX=J
   25 CONTINUE
      J=JMAX
      IF(J.GT.K) GOTO 31
      AATMP=ABS(ARRAY(K,K))
      IF (AATMP .GE. 1.0E-05) GOTO 41
   30 DETERM =0.
      GOTO 60
   31 DO 34 I=K,NORDER
      SAVE=ARRAY(I,J)
      ARRAY(I,J)=ARRAY(I,K)
   34 ARRAY(I,K)=SAVE
      DETERM=-DETERM
C**
C**       SUBTRACT ROW K FROM LOWER ROWS TO GET DIAGONAL MATRIX
C**
   41 DETERM=DETERM*ARRAY(K,K)
      IF(DETERM.EQ.0.0) RETURN
      IF(K-NORDER) 43,50,50
   43 K1=K+1
      DO 46 I=K1,NORDER
      DO 46 J=K1,NORDER
   46 ARRAY(I,J)=ARRAY(I,J)-ARRAY(I,K)*ARRAY(K,J)/ARRAY(K,K)
   50 CONTINUE
   60 RETURN
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AATMP     222B                             REAL
  AMAX      215B                             REAL
  ARRAY       1   DUMMY-ARG                  REAL      ADJ-ARY
  DETERM    212B                             REAL
  I         223B                             INTEGER
  J         217B                             INTEGER
  JMAX      216B                             INTEGER
  K         213B                             INTEGER
  K1        226B                             INTEGER
  NORDER      2   DUMMY-ARG                  INTEGER
  SAVE      225B                             REAL
  TMP       221B                             REAL


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ABS       GENERIC       1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    10 *NO REFS*                   19            34  INACTIVE   DO-TERM         42
    11 *NO REFS*                   20            41    124B                     47
    21 *NO REFS*                   24            43  INACTIVE                   50
    25     44B     DO-TERM         32            46  INACTIVE   DO-TERM         53
    30 *NO REFS*                   37            50    200B     DO-TERM         54
    31     71B                     39            60    205B                     55


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DETERM      6B     2


--STATISTICS--

  PROGRAM-UNIT LENGTH           233B =  155
  CM STORAGE USED             61000B =  25088
  COMPILE TIME                0.111 SECONDS
```

```
      SUBROUTINE ECALC                                                  ECALC      1
      DIMENSION PVECTOR(20)                                             ECALC      2
      REAL NUM                                                          ECALC      3
      LOGICAL TLOW                                                      ECALC      4
************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
************************************************************************COMR       3
      INTEGER RMAX                                                      COMR       4
      PARAMETER (RMAX = 20)                                             COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR       6
      INTEGER NROOMS                                                    COMR       7
      REAL ROOM                                                         COMR       8
************************************************************************COMR       9
************************************************************************COMR      10
*****************************************************************       COMJ       1
*                                                                       COMJ       2
* COMMON FOR EVALUATION OF ROOM MATRIX                                  COMJ       3
*                                                                       COMJ       4
*****************************************************************       COMJ       5
      COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME           COMJ       6
     +,SWR(RMAX,6),IDIR                                                 COMJ       7
      REAL TMAT ,ENERGY,POWER,SWR                                       COMJ       8
      LOGICAL FTIME                                                     COMJ       9
**************************************************                      ECALC      7
* CALCULATE THE ENERGY BALANCE IN THE ROOMS                             ECALC      8
**************************************************                      ECALC      9
*                                                                       ECALC     10
**************************************************                      ECALC     11
* CALCULATE THE DENOMINATOR TERM                                        ECALC     12
**************************************************                      ECALC     13
      CALL SETUP                                                        ECALC     14
      DENOM=DETERM(TMAT,NROOMS)                                         ECALC     15
      IF(DENOM) 100,50,100                                              ECALC     16
**************************************************                      ECALC     17
* ERROR # 4: DENOMINATOR = 0.                                           ECALC     18
**************************************************                      ECALC     19
   50 IERR = 4                                                          ECALC     20
      CALL ERROR(IERR)                                                  ECALC     21
      RETURN                                                            ECALC     22
**************************************************                      ECALC     23
* CALCULATE THE INPUT POWER VECTOR TO EACH ROOM                         ECALC     24
**************************************************                      ECALC     25
100   TLOW=.TRUE.                                                       ECALC     26
         DO 300 ICOL = 1,NROOMS                                         ECALC     27
         SUM = 0.0                                                      ECALC     28
              DO 200 IPWR = 1,6                                         ECALC     29
              IROW = IPWR + NROOMS                                      ECALC     30
  200         SUM = SUM + POWER(IPWR) * ROOM(IROW,ICOL)                 ECALC     31
         PVECTOR(ICOL) = - SUM                                          ECALC     32
300      IF (SUM.GT.(1.0E-06)) TLOW=.FALSE.                             ECALC     33
**************************************************                      ECALC     34
* CHECK IF INPUT POWER IS TOO LOW                                       ECALC     35
**************************************************                      ECALC     36
      IF (TLOW.NEQV..TRUE.) GOTO 350                                    ECALC     37
**************************************************                      ECALC     38
* INPUT TOO LOW                                                         ECALC     39
**************************************************                      ECALC     40
      DO 310 ICOL=1,NROOMS                                              ECALC     41
  310 ENERGY(ICOL)=1.0E-05                                              ECALC     42
      RETURN                                                            ECALC     43
**************************************************                      ECALC     44
* SET UP NUMERATORS                                                     ECALC     45
**************************************************                      ECALC     46
* RENEW TMATRIX                                                         ECALC     47
```

```
     65 ************************************************                      ECALC      48
     66   350 DO 500 ICOL = 1,NROOMS                                          ECALC      49
     67           CALL SETUP                                                  ECALC      50
     68 ************************************************                      ECALC      51
     69 * PUT PVECTOR INTO PROPER COLUMN                                      ECALC      52
     70 ************************************************                      ECALC      53
     71                DO 400 IROW = 1,NROOMS                                 ECALC      54
     72   400          TMAT(IROW,ICOL) = PVECTOR(IROW)                        ECALC      55
     73 ************************************************                      ECALC      56
     74 * NOW CALCULATE THE ENERGY FOR THE ROOM REPRESENTED                   ECALC      57
     75 * BY ICOL                                                             ECALC      58
     76 ************************************************                      ECALC      59
     77           NUM = DETERM(TMAT,NROOMS)                                   ECALC      60
     78           ENERGY(ICOL) = NUM/DENOM                                    ECALC      61
     79   500 CONTINUE                                                        ECALC      62
     80       RETURN                                                          ECALC      63
     81       END                                                             ECALC      64


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  DENOM      213B                            REAL
  ENERGY     620B  /MAT/                     REAL            20
  FTIME      652B  /MAT/                     LOGICAL
  ICOL       215B                            INTEGER
  IDIR      1043B  /MAT/                     INTEGER
  IERR       214B                            INTEGER
  IPWR       220B                            INTEGER
  IROW       222B                            INTEGER
  NROOMS    1244B  /ROOMN/                   INTEGER
  NUM        211B                            REAL
  POWER      644B  /MAT/                     REAL             6
  PVECTOR    165B                            REAL            20
  RAREA     1245B  /ROOMN/                   REAL            20
  ROOM         0B  /ROOMN/                   REAL           676
  SUM        217B                            REAL
  SWR        653B  /MAT/                     REAL           120
  TLOW       212B                            LOGICAL
  TMAT         0B  /MAT/                     REAL           400


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  RMAX    INTEGER                        20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  DETERM    REAL            2      FUNCTION
  ERROR                     1      SUBROUTINE
  SETUP                     0      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    50  INACTIVE                  37           310  INACTIVE   DO-TERM        59
   100    22B                     43           350    107B                    66
   200  INACTIVE   DO-TERM        48           400  INACTIVE   DO-TERM        72
   300  INACTIVE   DO-TERM        50           500  INACTIVE   DO-TERM        79
```

```
--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  ECALC       5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH              230B =  152
  CM LABELLED COMMON LENGTH       2335B =  1245
  CM STORAGE USED                61000B =  25088
  COMPILE TIME                   0.101 SECONDS
```

```
      SUBROUTINE PPWR                                                   PPWR       1
************************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF       2
************************************************************************COMF       3
      INTEGER FMAX                                                      COMF       4
      PARAMETER (FMAX = 50)                                             COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF       6
     $                 FTOT                                             COMF       7
      COMMON /INITILC/ BLDG                                             COMF       8
      CHARACTER * 5 BLDG                                                COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF      10
      INTEGER QUALITY, FERR, FTOT                                       COMF      11
************************************************************************COMF      12
************************************************************************COMF      13
************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
************************************************************************COMR       3
      INTEGER RMAX                                                      COMR       4
      PARAMETER (RMAX = 20)                                             COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR       6
      INTEGER NROOMS                                                    COMR       7
      REAL ROOM                                                         COMR       8
************************************************************************COMR       9
************************************************************************COMR      10
****************************************************************        COMJ       1
*                                                                       COMJ       2
* COMMON FOR EVALUATION OF ROOM MATRIX                                  COMJ       3
*                                                                       COMJ       4
****************************************************************        COMJ       5
      COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME           COMJ       6
     +,SWR(RMAX,6),IDIR                                                 COMJ       7
      REAL TMAT ,ENERGY,POWER,SWR                                       COMJ       8
      LOGICAL FTIME                                                     COMJ       9
      REAL DB                                                           PPWR       5
      WRITE(*,20) FREQ                                                  PPWR       6
   20 FORMAT (/"    POWER BY DIRECTION 1-6 AT A FREQUENCY OF",1PE10.3," PPWR       7
     +HZ")                                                              PPWR       8
      WRITE (*,30) (POWER(I),I=1,6)                                     PPWR       9
   30 FORMAT ( "         1         2         3         4         5      PPWR      10
     + 6", /," *********************************************************PPWR      11
     +*****",/, 6(3X,F7.2),/,)                                          PPWR      12
      WRITE (*,40)                                                      PPWR      13
   40 FORMAT("   ROOM          ENERGY         DB ",/,                   PPWR      14
     +  "*****************************************")                    PPWR      15
      DO 100 ICOL=1,NROOMS                                              PPWR      16
      DB=10.0 * ALOG10 ( ENERGY( ICOL) / 10. )                          PPWR      17
  100 WRITE(*,50) ICOL, ENERGY (ICOL), DB                               PPWR      18
   50 FORMAT( 3X,I3,5X,F10.2,5X,F10.2)                                  PPWR      19
      RETURN                                                            PPWR      20
      END                                                               PPWR      21
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG       2B  /INITILN/                  REAL
  BLDG        0B  /INITILC/                  CHAR*5
  DB        147B                             REAL
  ENERGY    620B  /MAT/                      REAL           20
  FERR       66B  /INITILN/                  INTEGER
  FREQ        0B  /INITILN/                  REAL
  FREQA       4B  /INITILN/                  REAL           50
  FTIME     652B  /MAT/                      LOGICAL
  FTOT       67B  /INITILN/                  INTEGER
```

```
 I          150B                                 INTEGER
 ICOL       151B                                 INTEGER
 IDIR      1043B  /MAT/                          INTEGER
 NROOMS    1244B  /ROOMN/                        INTEGER
 POWER      644B  /MAT/                          REAL             6
 QUALITY      1B  /INITILN/                      INTEGER
 RAREA     1245B  /ROOMN/                        REAL            20
 RFLAG        3B  /INITILN/                      REAL
 ROOM         0B  /ROOMN/                        REAL           676
 SWR        653B  /MAT/                          REAL           120
 TMAT         0B  /MAT/                          REAL           400


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX    INTEGER                         50
 RMAX    INTEGER                         20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 ALOG10    REAL           1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    20      55B     FORMAT         36
    30      65B     FORMAT         39
    40     105B     FORMAT         43
    50     117B     FORMAT         48
   100  INACTIVE    DO-TERM        47


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 PPWR        5B     0


--STATISTICS--

 PROGRAM-UNIT LENGTH            155B =  109
 CM LABELLED COMMON LENGTH     2426B =  1302
 CM STORAGE USED              61000B =  25088
 COMPILE TIME                 0.075 SECONDS
```

```
      SUBROUTINE PTMAT                                                  PTMAT      1
C** PRINTOUT THE CONTENTS OF THE ROOM MATRIX                             PTMAT      2
***********************************************************************COMF      1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF      2
***********************************************************************COMF      3
      INTEGER FMAX                                                      COMF      4
      PARAMETER (FMAX = 50)                                             COMF      5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF      6
     $                 FTOT                                             COMF      7
      COMMON /INITILC/ BLDG                                             COMF      8
      CHARACTER * 5 BLDG                                                COMF      9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF     10
      INTEGER QUALITY, FERR, FTOT                                       COMF     11
***********************************************************************COMF     12
***********************************************************************COMF     13
***********************************************************************COMR      1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR      2
***********************************************************************COMR      3
      INTEGER RMAX                                                      COMR      4
      PARAMETER (RMAX = 20)                                             COMR      5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR      6
      INTEGER NROOMS                                                    COMR      7
      REAL ROOM                                                         COMR      8
***********************************************************************COMR      9
***********************************************************************COMR     10
****************************************************************        COMJ      1
*                                                                       COMJ      2
* COMMON FOR EVALUATION OF ROOM MATRIX                                  COMJ      3
*                                                                       COMJ      4
****************************************************************        COMJ      5
      COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME           COMJ      6
     +,SWR(RMAX,6),IDIR                                                 COMJ      7
      REAL TMAT ,ENERGY,POWER,SWR                                       COMJ      8
      LOGICAL FTIME                                                     COMJ      9
      INTEGER R,C                                                       PTMAT      6
      PRINT*                                                            PTMAT      7
      PRINT*,'      TMAT MATRIX VALUES '                                PTMAT      8
      PRINT*,'  AT FREQUENCY = ',FREQ,' HERTZ'                          PTMAT      9
      PRINT*,'*********************************************************PTMAT     10
     +*************'                                                    PTMAT     11
      DO 10 R = 1,NROOMS                                                PTMAT     12
      PRINT 100,(TMAT(R,C), C = 1, NROOMS )                             PTMAT     13
   10 CONTINUE                                                          PTMAT     14
      PRINT*,'=========================================================PTMAT     15
     +============='                                                    PTMAT     16
  100 FORMAT(1X,12(E12.6) )                                             PTMAT     17
      RETURN                                                            PTMAT     18
      END                                                               PTMAT     19
```

--VARIABLE MAP--(LO=A)

-NAME---	ADDRESS--	BLOCK-----	PROPERTIES-------	TYPE---------	SIZE
AFLAG	2B	/INITILN/		REAL	
BLDG	0B	/INITILC/		CHAR*5	
C	146B			INTEGER	
ENERGY	620B	/MAT/		REAL	20
FERR	66B	/INITILN/		INTEGER	
FREQ	0B	/INITILN/		REAL	
FREQA	4B	/INITILN/		REAL	50
FTIME	652B	/MAT/		LOGICAL	
FTOT	67B	/INITILN/		INTEGER	
IDIR	1043B	/MAT/		INTEGER	
NROOMS	1244B	/ROOMN/		INTEGER	

```
 POWER      644B  /MAT/                      REAL           6
 QUALITY      1B  /INITILN/                  INTEGER
 R          145B                             INTEGER
 RAREA     1245B  /ROOMN/                    REAL          20
 RFLAG        3B  /INITILN/                  REAL
 ROOM         0B  /ROOMN/                    REAL         676
 SWR        653B  /MAT/                      REAL         120
 TMAT         0B  /MAT/                      REAL         400


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  FMAX    INTEGER                        50
  RMAX    INTEGER                        20


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        43
   100    113B     FORMAT         46


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  PTMAT       5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH            153B =  107
  CM LABELLED COMMON LENGTH     2426B =  1302
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.068 SECONDS
```

```
      SUBROUTINE DFACTOR                                                DFACTOR     1
************************************************************************COMF        1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF        2
************************************************************************COMF        3
      INTEGER FMAX                                                      COMF        4
      PARAMETER (FMAX = 50)                                             COMF        5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF        6
     $                  FTOT                                            COMF        7
      COMMON /INITILC/ BLDG                                             COMF        8
      CHARACTER * 5 BLDG                                                COMF        9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF       10
      INTEGER QUALITY, FERR, FTOT                                       COMF       11
************************************************************************COMF       12
************************************************************************COMF       13
************************************************************************COMH        1
***   COMMON FOR DATABASE OF LOCATIONS OF DOORS AND WINDOWS          ***COMH        2
************************************************************************COMH        3
      INTEGER HMAX                                                      COMH        4
      PARAMETER (HMAX = 35)                                             COMH        5
      COMMON /HOLEN/ HTOT, HERR                                         COMH        6
      COMMON /HOLEC/ HOLE(HMAX,4)                                       COMH        7
      INTEGER HTOT, HERR                                                COMH        8
      CHARACTER * 3 HOLE                                                COMH        9
* ==================================================                    COMH       10
* DESCRIPTION OF ARRAYS                                                 COMH       11
* ==================================================                    COMH       12
*     ROOM IDENTIFICATION            APERTURE ID                        COMH       13
* ------------------------------------                                  COMH       14
* DIRECTION    FROM ROOM     TO ROOM                                    COMH       15
* --------------------------------------------------                    COMH       16
* HOLE(X,1)    HOLE(X,2)     HOLE(X,3)    HOLE(X,4)                     COMH       17
*   A3            A3            A3            A3                        COMH       18
************************************************************************COMH       19
************************************************************************COMH       20
************************************************************************COMT        1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT        2
************************************************************************COMT        3
      INTEGER TMAX                                                      COMT        4
      PARAMETER (TMAX=35)                                               COMT        5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT        6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT        7
      INTEGER TTOT,TDBTOT,TERR                                          COMT        8
      REAL TDIM,TDB2                                                    COMT        9
      CHARACTER * 3 TYPE,TDB1                                           COMT       10
*================================================                       COMT       11
* DESCRIPTION OF ARRAYS                                                 COMT       12
*================================================                       COMT       13
*   ID        MATERIAL    FRAME MATERIAL                                COMT       14
*------------------------------------------------                       COMT       15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                      COMT       16
* A3           A3          A3                                           COMT       17
*================================================                       COMT       18
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT       19
*                         THICKNESS   ABOVE FLOOR                       COMT       20
*------------------------------------------------                       COMT       21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT       22
*  F8.2         F8.2        F8.2        F8.2                            COMT       23
*================================================                       COMT       24
*  ID       ATTENUATION    AREA                                         COMT       25
* -----------------------------------------------                       COMT       26
* TDB1(X)     TDB2(X,1)    TDB2(X,2)                                    COMT       27
*   A3          E9.3         E9.3                                       COMT       28
************************************************************************COMT       29
************************************************************************COMT       30
```

```
*************************************************************************COMR        1
***    COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS         ***COMR        2
*************************************************************************COMR        3
       INTEGER RMAX                                                      COMR        4
       PARAMETER (RMAX = 20)                                             COMR        5
       COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR        6
       INTEGER NROOMS                                                    COMR        7
       REAL ROOM                                                         COMR        8
*************************************************************************COMR        9
*************************************************************************COMR       10
*************************************************************************COMW        1
***    COMMON FOR DATABASE OF WALL PARAMETERS                        ***COMW        2
*************************************************************************COMW        3
       INTEGER WMAX                                                      COMW        4
       PARAMETER (WMAX = 75)                                             COMW        5
       COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW        6
       COMMON /WALLC/ WALL(WMAX,4)                                       COMW        7
       INTEGER WTOT,WERR                                                 COMW        8
       REAL WDIM                                                         COMW        9
       CHARACTER *3 WALL                                                 COMW       10
* =================================                                      COMW       11
** DESCRIPTION OF ARRAYS                                                 COMW       12
* =================================                                      COMW       13
*    WALL IDENTIFICATION                                                 COMW       14
* ---------------------------------                                      COMW       15
* DIRECTION    FROM        TO                                            COMW       16
*              ROOM        ROOM                                          COMW       17
* ---------------------------------                                      COMW       18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                      COMW       19
*  A3          A3          A3                                            COMW       20
* =====================================================                  COMW       21
*       WALL PARAMETERS                                                  COMW       22
* -----------------------------------------------------                  COMW       23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                    COMW       24
* -----------------------------------------------------                  COMW       25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                          COMW       26
*  A3          F8.2        F8.2        F8.2                              COMW       27
*************************************************************************COMW       28
*************************************************************************COMW       29
*************************************************************************COMD        1
*      COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS        COMD        2
*************************************************************************COMD        3
       COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ) ,DREFL, DREFLW           COMD        4
       REAL DDABS ,DREFL , DREFLW                                        COMD        5
*************************************************************************COMD        6
*************************************************************************COMD        7
       INTEGER NEXT, LAST, R, ROW                                        DFACTOR     8
       REAL AFACTOR,DREFLT                                               DFACTOR     9
       REAL WATTEN,LATTEN,OATTEN,MATTEN,ATTEN,T,S,TS,TS2                 DFACTOR    10
       REAL HEIGHT,WIDTH,AREA,OAREA,WAREA                                DFACTOR    11
       CHARACTER * 3 FROM, TO, MAT, ID                                   DFACTOR    12
       LOGICAL NEWWALL,WALLEND                                           DFACTOR    13
***                                                                      DFACTOR    14
*** THIS ROUTINE CALCULATES THE ABSORPTION OF THE WALL AND EACH          DFACTOR    15
*** OPENING IN THE WALL, LAYER BY LAYER, AND THEN CALCULATES THE         DFACTOR    16
*** COMPOSITE ABSORPTION BY WEIGHTING BY AREA EACH OPENING'S ABSORPTION DFACTOR    17
*** AND THE WALL ABSORPTION.                                             DFACTOR    18
*** FOR REFLECTIONS, THE ABSORPTION IS DECREASED BY "DREFL" IF           DFACTOR    19
*** THE WALL HAS A REFLECTION COEFFICIENT GREATER THAN 0.80 AND          DFACTOR    20
*** THE ROOM HAS A RESONANCE.                                            DFACTOR    21
***                                                                      DFACTOR    22
*** VARIABLE DEFINITIONS:                                                DFACTOR    23
***    DREFLT: EQUALS DREFL IF RESONANCE, OTHERWISE ZERO                 DFACTOR    24
***                 IT REPRESENTS REFLECTION GAINS                       DFACTOR    25
```

```
***   WATTEN: WALL ATTENUATION
***   OATTEN: OPENING ATTENUATION
***   LATTEN: LAYER ATTENUATION
***   MATTEN: MATERIAL ATTENUATION
***   MAT: MATERIAL IDENTIFICATION
***   WALL: WALL ARRAY CONTAINING WALL IDENTIFICATION AND MATERIAL
***   WDIM: WALL ARRAY CONTAINING PHYSICAL DIMENSIONS OF THE WALL
***   WMAX: MAXIMUM SIZE OF WALL AND WDIM ARRAYS
***   WTOT: TOTAL LINES OF DATA IN THE THE WALL AND WDIM ARRAYS.
***   HEIGHT: HEIGHT OF WALL
***   WIDTH: WIDTH OF WALL
***   T:       THICKNESS OF WALL
***   AREA: AREA
***   WAREA: TOTAL WALL AREA WITHOUT SUBTRACTING OPENINGS.
***   OAREA: TOTAL AREA OF THE OPENINGS.
***   NEWWALL: TRUE IF DATA LINE BELONGS TO A NEW WALL
***   WALLEND: TRUE IF DATA LINE IS THE LAST DATA LINE OF A WALL
***
      DO 10 R = 1,WTOT
***
*** SET WALLEND CONDITION
      NEXT = R + 1
      IF (R .EQ. WTOT) THEN
         WALLEND = .TRUE.
      ELSE IF ( WALL(R,2) .NE. WALL(NEXT,2) .OR.
     Z          WALL(R,3) .NE. WALL(NEXT,3) ) THEN
         WALLEND = .TRUE.
      ELSE
         WALLEND = .FALSE.
         END IF
***
*** SET NEWWALL CONDITION
      LAST = R - 1
      IF (R .EQ. 1) THEN
         NEWWALL = .TRUE.
      ELSE IF ( WALL(R,2) .NE. WALL(LAST,2) .OR.
     Z          WALL(R,3) .NE. WALL(LAST,3) ) THEN
         NEWWALL = .TRUE.
      ELSE
         NEWWALL = .FALSE.
         END IF
***
*** CALCULATE
      IF (NEWWALL) THEN
C*       ..INITIALIZE WALL CONDITIONS
      DREFLT = 0.0
         TS = 0
         TS2 = 0
         WATTEN = 0
         END IF
***   ..CALCULATE ATTENUATION FACTOR OF LAYER
      MAT = WALL(R,4)
      MATTEN = ATTEN (MAT,FREQ,AFLAG)
      CALL RESONW (WALL(R,2) , MAT)
      IF(DREFL.GT.0.0) DREFLT = DREFL
      LATTEN = MATTEN * WDIM(R,3)
***   ..CALCULATE RUNNING AFACTOR OF WALL
      WATTEN = WATTEN + LATTEN
      IF (WALLEND) THEN
         FROM = WALL(R,2)
         TO = WALL(R,3)
***   ...CALCULATE WEIGHTED AFACTOR OF OPENINGS
***   ...AND TOTAL AREA OF OPENINGS
         OAREA = 0
```

```
            DO 20 ROW = 1, HTOT
            IF (HOLE(ROW,2) .EQ. FROM .AND. HOLE(ROW,3) .EQ. TO) THEN
               ID = HOLE(ROW,4)
               CALL SRCHTDB(ID, OATTEN,AREA)
               OAREA = OAREA + AREA
               IF (OATTEN .LE. 120) THEN
                  T = 1.0 - 10**(-OATTEN / 10 )
               ELSE
                  T = 1.0
               ENDIF
               S = AREA
               TS = TS + T * S
               TS2 = TS2 + T * S * S
               END IF
 20         CONTINUE
 ***     ...CALCULATE TOTAL WALL AREA
            HEIGHT = WDIM(R,1)
            WIDTH = WDIM(R,2)
            WAREA = HEIGHT * WIDTH
            S = WAREA - OAREA
            IF (WATTEN .LE. 120.) THEN
               T = 1.0 - 10**(-WATTEN / 10) -DREFLT
            ELSE
               T = 1.0  - DREFLT
            ENDIF
            IF(T.LT.0.0) T=0.0
 ***     ...CALCULATE COMPOSITE ATTENUATION FACTOR OF WALL
            TS = TS + T * S
            TS2 = TS2 + T * S * S
 ***     ...INSERT COMPOSITE ATTENUATION OF WALL INTO ROOM MATRIX
            CALL LDDABS (TS,TS2,FROM,TO)
            END IF
 10      CONTINUE
         RETURN
         END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFACTOR  NONE                UNUSED/*S*        REAL
  AFLAG       2B  /INITILN/                      REAL
  AREA      446B                                 REAL
  BLDG        0B  /INITILC/                      CHAR*5
  DDABS       0B  /ROOMD/                        REAL            676
  DREFL    1244B  /ROOMD/                        REAL
  DREFLT    433B                                 REAL
  DREFLW   1245B  /ROOMD/                        REAL
  FERR       66B  /INITILN/                      INTEGER
  FREQ        0B  /INITILN/                      REAL
  FREQA       4B  /INITILN/                      REAL             50
  FROM      451B                                 CHAR*3
  FTOT       67B  /INITILN/                      INTEGER
  HEIGHT    444B                                 REAL
  HERR        1B  /HOLEN/                        INTEGER
  HOLE        0B  /HOLEC/                        CHAR*3          140
  HTOT        0B  /HOLEN/                        INTEGER
  ID        454B                                 CHAR*3
  LAST      430B                                 INTEGER
  LATTEN    435B                                 REAL
  MAT       453B                                 CHAR*3
  MATTEN    437B                                 REAL
  NEWWALL   455B                                 LOGICAL
  NEXT      427B                                 INTEGER
```

```
 NROOMS    1244B  /ROOMN/                        INTEGER
 OAREA      447B                                 REAL
 OATTEN     436B                                 REAL
 QUALITY      1B  /INITILN/                      INTEGER
 R          431B                                 INTEGER
 RAREA     1245B  /ROOMN/                        REAL            20
 RFLAG        3B  /INITILN/                      REAL
 ROOM         0B  /ROOMN/                        REAL           676
 ROW        432B                                 INTEGER
 S          441B                                 REAL
 T          440B                                 REAL
 TDBTOT     323B  /TYPEN/                        INTEGER
 TDB1        37B  /TYPEC/                        CHAR*3          35
 TDB2       215B  /TYPEN/                        REAL            70
 TDIM         0B  /TYPEN/                        REAL           140
 TERR       324B  /TYPEN/                        INTEGER
 TO         452B                                 CHAR*3
 TS         442B                                 REAL
 TS2        443B                                 REAL
 TTOT       214B  /TYPEN/                        INTEGER
 TYPE         0B  /TYPEC/                        CHAR*3         105
 WALL         0B  /WALLC/                        CHAR*3         300
 WALLEND    456B                                 LOGICAL
 WAREA      450B                                 REAL
 WATTEN     434B                                 REAL
 WDIM         0B  /WALLN/                        REAL           225
 WERR       342B  /WALLN/                        INTEGER
 WIDTH      445B                                 REAL
 WTOT       341B  /WALLN/                        INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX     INTEGER                        50
 HMAX     INTEGER                        35
 RMAX     INTEGER                        20
 TMAX     INTEGER                        35
 WMAX     INTEGER                        75


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 ATTEN      REAL           3      FUNCTION
 LDDABS                    4      SUBROUTINE
 RESONW                    2      SUBROUTINE
 SRCHTDB                   3      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM       225
    20  INACTIVE   DO-TERM       207
```

```
SUBROUTINE DFACTOR  74/175  OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DFACTOR      5B      0


--STATISTICS--

  PROGRAM-UNIT LENGTH              463B =   307
  CM LABELLED COMMON LENGTH       4000B =  2048
  CM STORAGE USED                63000B = 26112
  COMPILE TIME                   0.230 SECONDS
```

```
      SUBROUTINE LDDABS ( TS, TS2, FROM, TO )                           LDDABS      1
************************************************************************LDDABS      2
*   THIS ROUTINE LOADS THE ABSORPTION COEFFICIENT INTO THE APPROPRIATE  LDDABS      3
*   LOCATION IN THE 'DDABS' ARRAY.                                      LDDABS      4
*                                                                       LDDABS      5
*   NROOMS: TOTAL NUMBERS OF ROOMS REPRESENTED BY DATA                  LDDABS      6
*   RMAX: MAXIMUM NUMBER POSSIBLE UNDER THE PRESENT PROGRAM CONFIGURATIOLDDABS      7
*   TS AND TS2: ABSORPTION COEFFICIENTS                                 LDDABS      8
*   FROM: TO: CONTAINS ROOM#'S OR THE DIRECTIONS D1,D2,4,D5,OR D6.      LDDABS      9
************************************************************************LDDABS     10
************************************************************************COMR        1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR        2
************************************************************************COMR        3
      INTEGER RMAX                                                      COMR        4
      PARAMETER (RMAX = 20)                                             COMR        5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR        6
      INTEGER NROOMS                                                    COMR        7
      REAL ROOM                                                         COMR        8
************************************************************************COMR        9
************************************************************************COMR       10
************************************************************************COMD        1
*     COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS        COMD        2
************************************************************************COMD        3
      COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ) ,DREFL, DREFLW           COMD        4
      REAL DDABS ,DREFL , DREFLW                                        COMD        5
************************************************************************COMD        6
************************************************************************COMD        7
      INTEGER VAL, C, R, RNUM, D                                        LDDABS     13
      REAL TS, TS2                                                      LDDABS     14
      CHARACTER * 3 FROM, TO                                            LDDABS     15
************************************************************************LDDABS     16
      IF ( FROM(1:1) .EQ. 'D' ) THEN                                    LDDABS     17
         RNUM = VAL ( TO(1:2) )                                         LDDABS     18
         D = VAL ( FROM(2:2) )                                          LDDABS     19
************************************************************************LDDABS     20
*   INSERT ABSORPTION COEFFICENT FOR ENERGY ENTERING A ROOM FROM THE    LDDABS     21
*   OUTSIDE OF THE BUILDING.                                            LDDABS     22
************************************************************************LDDABS     23
         R = NROOMS + D                                                 LDDABS     24
         C = RNUM                                                       LDDABS     25
         DDABS(R,C) = TS  + DDABS(R,C)                                  LDDABS     26
************************************************************************LDDABS     27
*   INSERT ABSORPTION COEFFICIENT INTO 'DDABS' ARRAY FOR ENERGY LEAVING LDDABS     28
*   A ROOM TO THE OUTSIDE OF THE BUILDING.                              LDDABS     29
************************************************************************LDDABS     30
         R = RNUM                                                       LDDABS     31
         C = NROOMS + D                                                 LDDABS     32
         DDABS(R,C) = TS2 / RAREA(RNUM) + DDABS(R,C)                    LDDABS     33
************************************************************************LDDABS     34
      ELSE IF ( TO(1:1) .EQ. 'D' ) THEN                                 LDDABS     35
         RNUM = VAL ( FROM(1:2) )                                       LDDABS     36
         D = VAL ( TO(2:2) )                                            LDDABS     37
************************************************************************LDDABS     38
*   INSERT ABSORPTION COEFFICIENT INTO 'DDABS' ARRAY FOR ENERGY ENTERINGLDDABS     39
*   A ROOM FROM THE OUTSIDE OF THE BUILDING.                            LDDABS     40
************************************************************************LDDABS     41
         R = NROOMS + D                                                 LDDABS     42
         C = RNUM                                                       LDDABS     43
         DDABS(R,C) = TS  + DDABS(R,C)                                  LDDABS     44
************************************************************************LDDABS     45
*   INSERT ABSORPTION COEFFICIENT INTO 'DDABS' ARRAY FOR ENERGY LEAVING LDDABS     46
*   A ROOM TO THE OUTSIDE OF THE BUILDING.                              LDDABS     47
************************************************************************LDDABS     48
         R = RNUM                                                       LDDABS     49
```

```
              C = NROOMS + D                                                  LDDABS
              DDABS(R,C) = TS2 / RAREA(RNUM) + DDABS(R,C)                     LDDABS
      ************************************************************************LDDABS
           ELSE                                                               LDDABS
      ************************************************************************LDDABS
      *   INSERT ABSORPTION COEFFICIENTS INTO 'DDABS' ARRAY FOR ENERGY GOING  LDDABS
      *   FROM ROOM TO ROOM.                                                  LDDABS
      ************************************************************************LDDABS
              R = VAL ( FROM(1:2) )                                           LDDABS
              C = VAL ( TO(1:2) )                                             LDDABS
              DDABS(R,C) = TS2 / RAREA(R)  + DDABS(R,C)                       LDDABS
              DDABS(C,R) = TS2 / RAREA(C)  + DDABS(C,R)                       LDDABS
           ENDIF                                                              LDDABS
           RETURN                                                             LDDABS
           END                                                                LDDABS
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 C          222B                             INTEGER
 D          225B                             INTEGER
 DDABS        0B  /ROOMD/                    REAL           676
 DREFL     1244B  /ROOMD/                    REAL
 DREFLW    1245B  /ROOMD/                    REAL
 FROM         3   DUMMY-ARG                  CHAR*3
 NROOMS    1244B  /ROOMN/                    INTEGER
 R          223B                             INTEGER
 RAREA     1245B  /ROOMN/                    REAL            20
 RNUM       224B                             INTEGER
 ROOM         0B  /ROOMN/                    REAL           676
 TO           4   DUMMY-ARG                  CHAR*3
 TS           1   DUMMY-ARG                  REAL
 TS2          2   DUMMY-ARG                  REAL


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 RMAX     INTEGER                        20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 VAL        INTEGER         1       FUNCTION


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 LDDABS       5B      4


--STATISTICS--

 PROGRAM-UNIT LENGTH              230B =   152
 CM LABELLED COMMON LENGTH       2537B =  1375
 CM STORAGE USED                61000B = 25088
 COMPILE TIME                   0.103 SECONDS
```

```
      SUBROUTINE PDDABS                                                 PDDABS   1
C** PRINTOUT THE CONTENTS OF THE ROOM MATRIX                            PDDABS   2
************************************************************************COMF     1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF     2
************************************************************************COMF     3
      INTEGER FMAX                                                      COMF     4
      PARAMETER (FMAX = 50)                                             COMF     5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF     6
     $                 FTOT                                             COMF     7
      COMMON /INITILC/ BLDG                                             COMF     8
      CHARACTER * 5 BLDG                                                COMF     9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF    10
      INTEGER QUALITY, FERR, FTOT                                       COMF    11
************************************************************************COMF    12
************************************************************************COMF    13
************************************************************************COMR     1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR     2
************************************************************************COMR     3
      INTEGER RMAX                                                      COMR     4
      PARAMETER (RMAX = 20)                                             COMR     5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR     6
      INTEGER NROOMS                                                    COMR     7
      REAL ROOM                                                         COMR     8
************************************************************************COMR     9
************************************************************************COMR    10
*****************************************************************       COMJ     1
*                                                                       COMJ     2
* COMMON FOR EVALUATION OF ROOM MATRIX                                  COMJ     3
*                                                                       COMJ     4
*****************************************************************       COMJ     5
      COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME           COMJ     6
     +,SWR(RMAX,6),IDIR                                                 COMJ     7
      REAL TMAT ,ENERGY,POWER,SWR                                       COMJ     8
      LOGICAL FTIME                                                     COMJ     9
************************************************************************COMD     1
*     COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS        COMD     2
************************************************************************COMD     3
      COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ) ,DREFL, DREFLW           COMD     4
      REAL DDABS ,DREFL , DREFLW                                        COMD     5
************************************************************************COMD     6
************************************************************************COMD     7
      INTEGER R,C                                                       PDDABS   7
      PRINT*                                                            PDDABS   8
      PRINT*,'     DDABS MATRIX VALUES '                                PDDABS   9
      PRINT*,'  AT FREQUENCY = ',FREQ,' HERTZ'                          PDDABS  10
      PRINT*,'  WITH AFLAG = ', AFLAG, ' PER CENT'                      PDDABS  11
      PRINT*,'**********************************************************PDDABS  12
     +*************'                                                    PDDABS  13
      DO 10 R = 1,NROOMS + 6                                            PDDABS  14
      PRINT 100,(DDABS(R,C), C = 1, NROOMS + 6 )                        PDDABS  15
   10 CONTINUE                                                          PDDABS  16
      PRINT*,'==========================================================PDDABS  17
     +============='                                                    PDDABS  18
  100 FORMAT(1X,12(F8.3) )                                              PDDABS  19
      RETURN                                                            PDDABS  20
      END                                                               PDDABS  21
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 AFLAG       2B  /INITILN/                  REAL
 BLDG        0B  /INITILC/                  CHAR*5
 C         163B                             INTEGER
 DDABS       0B  /ROOMD/                    REAL              676
 DREFL    1244B  /ROOMD/                    REAL
 DREFLW   1245B  /ROOMD/                    REAL
 ENERGY    620B  /MAT/                      REAL               20
 FERR       66B  /INITILN/                  INTEGER
 FREQ        0B  /INITILN/                  REAL
 FREQA       4B  /INITILN/                  REAL               50
 FTIME     652B  /MAT/                      LOGICAL
 FTOT       67B  /INITILN/                  INTEGER
 IDIR     1043B  /MAT/                      INTEGER
 NROOMS   1244B  /ROOMN/                    INTEGER
 POWER     644B  /MAT/                      REAL                6
 QUALITY     1B  /INITILN/                  INTEGER
 R         162B                             INTEGER
 RAREA    1245B  /ROOMN/                    REAL               20
 RFLAG       3B  /INITILN/                  REAL
 ROOM        0B  /ROOMN/                    REAL              676
 SWR       653B  /MAT/                      REAL              120
 TMAT        0B  /MAT/                      REAL              400


--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE-------------------------VALUE

 FMAX    INTEGER                        50
 RMAX    INTEGER                        20


--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        51
   100     121B    FORMAT         54


--ENTRY POINTS--(LO=A)
-NAME---ADDRESS--ARGS---

 PDDABS      5B     0


--STATISTICS--

 PROGRAM-UNIT LENGTH              170B =   120
 CM LABELLED COMMON LENGTH       3674B =  1980
 CM STORAGE USED                61000B = 25088
 COMPILE TIME                   0.071 SECONDS
```

```
 1        SUBROUTINE IDDABS                                                  IDDABS     1
 2 *******************************************************************************IDDABS     2
 3 *   INITIALIZE DDABS MATRIX.                                              IDDABS     3
 4 *******************************************************************************IDDABS     4
 5 *******************************************************************************COMR       1
 6 ***    COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
 7 *******************************************************************************COMR       3
 8        INTEGER RMAX                                                       COMR       4
 9        PARAMETER (RMAX = 20)                                              COMR       5
10        COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)       COMR       6
11        INTEGER NROOMS                                                     COMR       7
12        REAL ROOM                                                          COMR       8
13 *******************************************************************************COMR       9
14 *******************************************************************************COMR      10
15 *******************************************************************************COMD       1
16 *      COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS        COMD       2
17 *******************************************************************************COMD       3
18        COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ) ,DREFL, DREFLW            COMD       4
19        REAL DDABS ,DREFL , DREFLW                                         COMD       5
20 *******************************************************************************COMD       6
21 *******************************************************************************COMD       7
22        INTEGER R,C                                                        IDDABS     7
23        DO 10 R = 1,RMAX                                                   IDDABS     8
24        DO 10 C = 1,RMAX                                                   IDDABS     9
25        DDABS(R,C) = 0.0                                                   IDDABS    10
26     10 CONTINUE                                                           IDDABS    11
27        DO 20 R = 1,RMAX                                                   IDDABS    12
28        DO 20 C = RMAX + 1, RMAX + 5                                       IDDABS    13
29        DDABS(R,C) = 0.0                                                   IDDABS    14
30     20 CONTINUE                                                           IDDABS    15
31        RETURN                                                             IDDABS    16
32        END                                                                IDDABS    17
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 C          70B                             INTEGER
 DDABS       0B  /ROOMD/                    REAL          676
 DREFL    1244B  /ROOMD/                    REAL
 DREFLW   1245B  /ROOMD/                    REAL
 NROOMS   1244B  /ROOMN/                    INTEGER
 R          67B                             INTEGER
 RAREA    1245B  /ROOMN/                    REAL           20
 ROOM        0B  /ROOMN/                    REAL          676


--SYMBOLIC CONSTANTS--(LO=A)
-NAME----TYPE------------------------VALUE

 RMAX    INTEGER                        20


--STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        26
    20  INACTIVE   DO-TERM        30
```

```
 SUBROUTINE IDDABS   74/175  OPT=0

--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  IDDABS      5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH              77B =  63
  CM LABELLED COMMON LENGTH      2537B =  1375
  CM STORAGE USED               61000B =  25088
  COMPILE TIME                  0.050 SECONDS
```

```
      SUBROUTINE PPWR2                                                  PPWR2         1
************************************************************************COMF          1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF          2
************************************************************************COMF          3
      INTEGER FMAX                                                      COMF          4
      PARAMETER (FMAX = 50)                                             COMF          5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF          6
     $                  FTOT                                            COMF          7
      COMMON /INITILC/ BLDG                                             COMF          8
      CHARACTER * 5 BLDG                                                COMF          9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF         10
      INTEGER QUALITY, FERR, FTOT                                       COMF         11
************************************************************************COMF         12
************************************************************************COMF         13
************************************************************************COMR          1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR          2
************************************************************************COMR          3
      INTEGER RMAX                                                      COMR          4
      PARAMETER (RMAX = 20)                                             COMR          5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR          6
      INTEGER NROOMS                                                    COMR          7
      REAL ROOM                                                         COMR          8
************************************************************************COMR          9
************************************************************************COMR         10
*****************************************************************       COMJ          1
*                                                                       COMJ          2
* COMMON FOR EVALUATION OF ROOM MATRIX                                  COMJ          3
*                                                                       COMJ          4
*****************************************************************       COMJ          5
      COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME           COMJ          6
     +,SWR(RMAX,6),IDIR                                                 COMJ          7
      REAL TMAT ,ENERGY,POWER,SWR                                       COMJ          8
      LOGICAL FTIME                                                     COMJ          9
      REAL DB                                                           PPWR2         5
      WRITE(*,30) FREQ                                                  PPWR2         6
   30 FORMAT (//,"ATTENUATION AT A FREQUENCY OF",1PE10.3," HZ"          PPWR2         7
     +,/,"**************************************************",          PPWR2         8
     +"****************************"                                    PPWR2         9
     +,/,"*        *                         DIRECTIONS   ",28X,"*",/,  PPWR2        10
     +"* ROOMS *      1             2             3             4  "  , PPWR2        11
     +"           5    *",/,"********************************************" ,PPWR2  12
     +"**********************************")                             PPWR2        13
      WRITE(*,100) (IROW,(SWR(IROW,I),I=1,5), IROW=1,NROOMS)            PPWR2        14
  100 FORMAT ("* ",I3,3X,"*",F9.2,3X,F10.2,3X,F10.2,3X,F10.2,3X,F10.2,  PPWR2        15
     +"  *")                                                            PPWR2        16
      WRITE (*,120)                                                     PPWR2        17
  120 FORMAT ("**************************************************" ,     PPWR2        18
     +"****************************")                                   PPWR2        19
      RETURN                                                            PPWR2        20
      END                                                               PPWR2        21
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 AFLAG       2B  /INITILN/                  REAL
 BLDG        0B  /INITILC/                  CHAR*5
 DB        NONE              UNUSED/*S*     REAL
 ENERGY    620B  /MAT/                      REAL            20
 FERR       66B  /INITILN/                  INTEGER
 FREQ        0B  /INITILN/                  REAL
 FREQA       4B  /INITILN/                  REAL            50
 FTIME     652B  /MAT/                      LOGICAL
 FTOT       67B  /INITILN/                  INTEGER
```

```
 I          167B                         INTEGER
 IDIR      1043B  /MAT/                  INTEGER
 IROW       165B                         INTEGER
 NROOMS    1244B  /ROOMN/                INTEGER
 POWER      644B  /MAT/                  REAL             6
 QUALITY      1B  /INITILN/              INTEGER
 RAREA     1245B  /ROOMN/                REAL            20
 RFLAG        3B  /INITILN/              REAL
 ROOM         0B  /ROOMN/                REAL           676
 SWR        653B  /MAT/                  REAL           120
 TMAT         0B  /MAT/                  REAL           400


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX    INTEGER                         50
 RMAX    INTEGER                         20


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    30     56B     FORMAT          36
   100    123B     FORMAT          44
   120    133B     FORMAT          47


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 PPWR2       5B     0


--STATISTICS--

 PROGRAM-UNIT LENGTH            173B =  123
 CM LABELLED COMMON LENGTH     2426B =  1302
 CM STORAGE USED              61000B =  25088
 COMPILE TIME                 0.071 SECONDS
```

```
      SUBROUTINE SPWR                                                   SPWR       1
************************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF       2
************************************************************************COMF       3
      INTEGER FMAX                                                      COMF       4
      PARAMETER (FMAX = 50)                                             COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF       6
     $                  FTOT                                            COMF       7
      COMMON /INITILC/ BLDG                                             COMF       8
      CHARACTER * 5 BLDG                                                COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF      10
      INTEGER QUALITY, FERR, FTOT                                       COMF      11
************************************************************************COMF      12
************************************************************************COMF      13
************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
************************************************************************COMR       3
      INTEGER RMAX                                                      COMR       4
      PARAMETER (RMAX = 20)                                             COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR       6
      INTEGER NROOMS                                                    COMR       7
      REAL ROOM                                                         COMR       8
************************************************************************COMR       9
************************************************************************COMR      10
****************************************************************        COMJ       1
*                                                                       COMJ       2
* COMMON FOR EVALUATION OF ROOM MATRIX                                  COMJ       3
*                                                                       COMJ       4
****************************************************************        COMJ       5
      COMMON /MAT/TMAT(RMAX,RMAX),ENERGY(RMAX),POWER(6),FTIME           COMJ       6
     +,SWR(RMAX,6),IDIR                                                 COMJ       7
      REAL TMAT ,ENERGY,POWER,SWR                                       COMJ       8
      LOGICAL FTIME                                                     COMJ       9
      REAL DB                                                           SPWR       5
      DO 100 IROW=1,NROOMS                                              SPWR       6
      IF(ENERGY(IROW).LT.1.0E-05) THEN                                  SPWR       7
      DB = -60.0                                                        SPWR       8
      GO TO 100                                                         SPWR       9
      ENDIF                                                             SPWR      10
        DB=10.0 * ALOG10 ( ENERGY( IROW) / 10. )                        SPWR      11
  100 SWR(IROW,IDIR)=DB                                                 SPWR      12
      RETURN                                                            SPWR      13
      END                                                               SPWR      14
```

```
--VARIABLE MAP--(LO=A)
-NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 AFLAG       2B  /INITILN/                  REAL
 BLDG        0B  /INITILC/                  CHAR*5
 DB         55B                             REAL
 ENERGY    620B  /MAT/                      REAL            20
 FERR       66B  /INITILN/                  INTEGER
 FREQ        0B  /INITILN/                  REAL
 FREQA       4B  /INITILN/                  REAL            50
 FTIME     652B  /MAT/                      LOGICAL
 FTOT       67B  /INITILN/                  INTEGER
 IDIR     1043B  /MAT/                      INTEGER
 IROW       56B                             INTEGER
 NROOMS   1244B  /ROOMN/                    INTEGER
 POWER     644B  /MAT/                      REAL             6
 QUALITY     1B  /INITILN/                  INTEGER
 RAREA    1245B  /ROOMN/                    REAL            20
 RFLAG       3B  /INITILN/                  REAL
```

```
 ROOM        0B  /ROOMN/                     REAL          676
 SWR       653B  /MAT/                       REAL          120
 TMAT        0B  /MAT/                       REAL          400


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

  FMAX    INTEGER                         50
  RMAX    INTEGER                         20


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  ALOG10    REAL           1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    100     31B     DO-TERM        41


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  SPWR        5B     0


--STATISTICS--

  PROGRAM-UNIT LENGTH             62B =  50
  CM LABELLED COMMON LENGTH     2426B =  1302
  CM STORAGE USED              61000B =  25088
  COMPILE TIME                 0.057 SECONDS
```

```
      SUBROUTINE RESONW(FROM,MATID)                                     RESONW     1
*                                                                       RESONW     2
*         THIS ROUTINE CHECKS FOR RESONANCE CONDITIONS IN A ROOM        RESONW     3
*         IF A RESONANCE IS POSSIBLE AT THE PARTICULAR FREQUENCY        RESONW     4
*         FOR THE ROOM AND THE WALL IN QUESTION HAS A REFLECTION        RESONW     5
*         COEFFICIENT GREATER THAN 0.80, THEN THE ABSORPTION FOR        RESONW     6
*         THE WALL IS REDUCED BY THE REFLECTION COEFFICIENT.            RESONW     7
*                                                                       RESONW     8
************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
************************************************************************COMR       3
      INTEGER RMAX                                                      COMR       4
      PARAMETER (RMAX = 20)                                             COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR       6
      INTEGER NROOMS                                                    COMR       7
      REAL ROOM                                                         COMR       8
************************************************************************COMR       9
************************************************************************COMR      10
************************************************************************COMW       1
***   COMMON FOR DATABASE OF WALL PARAMETERS                         ***COMW       2
************************************************************************COMW       3
      INTEGER WMAX                                                      COMW       4
      PARAMETER (WMAX = 75)                                             COMW       5
      COMMON /WALLN/ WDIM(WMAX,3), WTOT, WERR                           COMW       6
      COMMON /WALLC/ WALL(WMAX,4)                                       COMW       7
      INTEGER WTOT,WERR                                                 COMW       8
      REAL WDIM                                                         COMW       9
      CHARACTER *3 WALL                                                 COMW      10
* ================================                                      COMW      11
** DESCRIPTION OF ARRAYS                                                COMW      12
* ================================                                      COMW      13
*    WALL IDENTIFICATION                                                COMW      14
* --------------------------------                                      COMW      15
* DIRECTION    FROM       TO                                            COMW      16
*              ROOM       ROOM                                          COMW      17
* --------------------------------                                      COMW      18
* WALL(X,1)   WALL(X,2)   WALL(X,3)                                     COMW      19
*  A3          A3          A3                                           COMW      20
* ====================================================                  COMW      21
*        WALL PARAMETERS                                                COMW      22
* ----------------------------------------------------                  COMW      23
* MATERIAL    HEIGHT      WIDTH       LAYER THICKNESS                   COMW      24
* ----------------------------------------------------                  COMW      25
* WALL(X,4)   WDIM(X,1)   WDIM(X,2)   WDIM(X,3)                         COMW      26
*  A3          F8.2        F8.2        F8.2                             COMW      27
************************************************************************COMW      28
************************************************************************COMW      29
************************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF       2
************************************************************************COMF       3
      INTEGER FMAX                                                      COMF       4
      PARAMETER (FMAX = 50)                                             COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF       6
     $                 FTOT                                             COMF       7
      COMMON /INITILC/ BLDG                                             COMF       8
      CHARACTER * 5 BLDG                                                COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF      10
      INTEGER QUALITY, FERR, FTOT                                       COMF      11
************************************************************************COMF      12
************************************************************************COMF      13
************************************************************************COMM       1
***   COMMON FOR DATABASE OF MATERIAL PROPERTIES                     ***COMM       2
************************************************************************COMM       3
      INTEGER MMAX                                                      COMM       4
```

```
      PARAMETER (MMAX=100)                                              COMM       5
      COMMON /MATN/ MATTEN(MMAX,7), MRCOEF(MMAX,7), QA(MMAX), QR(MMAX), COMM       6
     $               MFREQ(MMAX,7), MERR, MTOT                          COMM       7
      COMMON /MATC/MAT(MMAX),MATDESC(MMAX)                              COMM       8
      INTEGER MTOT, MERR                                                COMM       9
      REAL MATTEN, MRCOEF, MFREQ, QA, QR                                COMM      10
      CHARACTER * 3  MAT                                                COMM      11
      CHARACTER * 70 MATDESC                                            COMM      12
************************************************************************COMM      13
************************************************************************COMM      14
************************************************************************COMD       1
*     COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS        COMD       2
************************************************************************COMD       3
      COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ) ,DREFL, DREFLW           COMD       4
      REAL DDABS ,DREFL , DREFLW                                        COMD       5
************************************************************************COMD       6
************************************************************************COMD       7
      REAL A,B,C,RH,RL,RW                                               RESONW    14
      REAL MREFL                                                        RESONW    15
      CHARACTER * 3 FROM,MATID                                          RESONW    16
      ISET = 0                                                          RESONW    17
      MREFL = RCOEF (MATID,FREQ,RFLAG)                                  RESONW    18
      IF(MREFL.LT.0.80) THEN                                            RESONW    19
      DREFL = 0.0                                                       RESONW    20
      RETURN                                                            RESONW    21
      ENDIF                                                             RESONW    22
      IF(FROM(1:1) .EQ. 'D' ) THEN                                      RESONW    23
      DREFL = 0.0                                                       RESONW    24
      RETURN                                                            RESONW    25
      ENDIF                                                             RESONW    26
*                                                                       RESONW    27
*        NOW GET HEIGHT, WIDTH AND LENGTH OF ROOM                       RESONW    28
*                                                                       RESONW    29
      DO 100 I1 = 1, WTOT                                               RESONW    30
      IF( FROM.EQ.WALL(I1,2) .AND. WALL(I1,1).EQ.'FB ') THEN            RESONW    31
      RH = WDIM(I1,1)                                                   RESONW    32
      GOTO 200                                                          RESONW    33
      ENDIF                                                             RESONW    34
  100 CONTINUE                                                          RESONW    35
      DREFL = 0                                                         RESONW    36
      IWARN = 9                                                         RESONW    37
      CALL WARNING(IWARN)                                               RESONW    38
      RETURN                                                            RESONW    39
  200 I1 = 0                                                            RESONW    40
  300 I1 = I1 + 1                                                       RESONW    41
      IF(I1.GT.WTOT) THEN                                               RESONW    42
      DREFL = 0.0                                                       RESONW    43
      IF(ISET.EQ.1) RETURN                                              RESONW    44
      IWARN = 10                                                        RESONW    45
      CALL WARNING(IWARN)                                               RESONW    46
      RETURN                                                            RESONW    47
      ENDIF                                                             RESONW    48
      IF(FROM.EQ.WALL(I1,2) .AND. WALL(I1,1).EQ.'UD ') THEN             RESONW    49
      RL= WDIM(I1,2)                                                    RESONW    50
      RW= WDIM(I1,1)                                                    RESONW    51
      ELSE                                                              RESONW    52
      GOTO 300                                                          RESONW    53
      ENDIF                                                             RESONW    54
*                                                                       RESONW    55
*        NOW SORT OUT DIMENSIONS WITH A SMALLEST AND C LARGEST           RESONW    56
      ISET = 1                                                          RESONW    57
      IPASS = 0                                                         RESONW    58
      A = RH                                                            RESONW    59
      B = RW                                                            RESONW    60
```

```
      C = RL
 500  IF(A.GT.B) THEN
      TMP= A
      A = B
      B = TMP
      IPASS = 0
      ELSE
      IPASS = 1
      ENDIF
      IF (B.GT.C) THEN
      TMP =B
      B = C
      C = TMP
      IPASS = 0
      ENDIF
      IF (IPASS.EQ. 0) GOTO 500
*
*        NOW CALCULATE LOWER RESONANCE FREQUENCY
      CLIGHT= 3.0E08
      FLOW = 1.0/(B*B)  + 1.0/ (C*C)
      FLOW = SQRT(FLOW)
      FLOW = FLOW*CLIGHT/2.0
*
*        NOW CALCULATE HIGH FREQUENCY LIMIT
      FHIGH = 9.0*( 1.0/(A*A) + 1.0/(B*B) + 1.0/(C*C) )
      FHIGH = SQRT (FHIGH)
      FHIGH = FHIGH*CLIGHT/2.0
      IF(FREQ.GE.FLOW .AND. FREQ.LE.FHIGH) THEN
      DREFL = MREFL
      RETURN
      ELSE
      DREFL = 0.0
      GOTO 300
      ENDIF
      END
```

--VARIABLE MAP--(LO=A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
A	346B			REAL	
AFLAG	2B	/INITILN/		REAL	
B	347B			REAL	
BLDG	0B	/INITILC/		CHAR*5	
C	350B			REAL	
CLIGHT	363B			REAL	
DDABS	0B	/ROOMD/		REAL	676
DREFL	1244B	/ROOMD/		REAL	
DREFLW	1245B	/ROOMD/		REAL	
FERR	66B	/INITILN/		INTEGER	
FHIGH	365B			REAL	
FLOW	364B			REAL	
FREQ	0B	/INITILN/		REAL	
FREQA	4B	/INITILN/		REAL	50
FROM	1	DUMMY-ARG		CHAR*3	
FTOT	67B	/INITILN/		INTEGER	
IPASS	361B			INTEGER	
ISET	355B			INTEGER	
IWARN	360B			INTEGER	
I1	356B			INTEGER	
MAT	0B	/MATC/		CHAR*3	100
MATDESC	36B	/MATC/		CHAR*70	100
MATID	2	DUMMY-ARG		CHAR*3	
MATTEN	0B	/MATN/		REAL	700

```
 MERR      4374B  /MATN/                  INTEGER
 MFREQ     3100B  /MATN/                  REAL           700
 MRCOEF    1274B  /MATN/                  REAL           700
 MREFL      354B                          REAL
 MTOT      4375B  /MATN/                  INTEGER
 NROOMS    1244B  /ROOMN/                 INTEGER
 QA        2570B  /MATN/                  REAL           100
 QR        2734B  /MATN/                  REAL           100
 QUALITY      1B  /INITILN/               INTEGER
 RAREA     1245B  /ROOMN/                 REAL            20
 RFLAG        3B  /INITILN/               REAL
 RH         351B                          REAL
 RL         352B                          REAL
 ROOM         0B  /ROOMN/                 REAL           676
 RW         353B                          REAL
 TMP        362B                          REAL
 WALL         0B  /WALLC/                 CHAR*3         300
 WDIM         0B  /WALLN/                 REAL           225
 WERR       342B  /WALLN/                 INTEGER
 WTOT       341B  /WALLN/                 INTEGER


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX     INTEGER                        50
 MMAX     INTEGER                       100
 RMAX     INTEGER                        20
 WMAX     INTEGER                        75


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 RCOEF      REAL           3      FUNCTION
 SQRT       GENERIC        1      INTRINSIC
 WARNING                   1      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

   100  INACTIVE   DO-TERM       103
   200    102B                   108
   300    104B                   109
   500    162B                   130


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 RESONW      5B     2


--STATISTICS--

 PROGRAM-UNIT LENGTH           370B =   248
 CM LABELLED COMMON LENGTH   11255B =  4781
 CM STORAGE USED             61000B = 25088
 COMPILE TIME                0.190 SECONDS
```

```
      SUBROUTINE RESOND(ID)                                             RESOND     1
*                                                                       RESOND     2
*                                                                       RESOND     3
*          THIS ROUTINE CALCULATES RESONANCES FOR DOORS AND             RESOND     4
*          WINDOWS.  IN THE RESONANCE FREQUENCY RANGE,  THE             RESOND     5
*          INPUT TRANSMISSION OF THE WINDOW OR DOOR IS  IN-             RESOND     6
*          BY 20 DB (ARBITRARY).                                        RESOND     7
*                                                                       RESOND     8
*                                                                       RESOND     9
************************************************************************COMR       1
***   COMMON FOR ROOM ARRAY CONTAINING ATTENUATION OF WALLS          ***COMR       2
************************************************************************COMR       3
      INTEGER RMAX                                                      COMR       4
      PARAMETER (RMAX = 20)                                             COMR       5
      COMMON /ROOMN/ ROOM(RMAX + 6, RMAX + 6), NROOMS, RAREA(RMAX)      COMR       6
      INTEGER NROOMS                                                    COMR       7
      REAL ROOM                                                         COMR       8
************************************************************************COMR       9
************************************************************************COMR      10
************************************************************************COMD       1
*     COMMON FOR ABSORPTION AND REFLECTION COEFFICIENTS IN WALLS        COMD       2
************************************************************************COMD       3
      COMMON /ROOMD/DDABS(RMAX + 6, RMAX + 6 ) ,DREFL, DREFLW           COMD       4
      REAL DDABS ,DREFL , DREFLW                                        COMD       5
************************************************************************COMD       6
************************************************************************COMD       7
************************************************************************COMF       1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF       2
************************************************************************COMF       3
      INTEGER FMAX                                                      COMF       4
      PARAMETER (FMAX = 50)                                             COMF       5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF       6
     $                 FTOT                                             COMF       7
      COMMON /INITILC/ BLDG                                             COMF       8
      CHARACTER * 5 BLDG                                                COMF       9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF      10
      INTEGER QUALITY, FERR, FTOT                                       COMF      11
************************************************************************COMF      12
************************************************************************COMF      13
************************************************************************COMT       1
***   COMMON FOR DATABASE OF TYPES OF DOORS AND WINDOWS              ***COMT       2
************************************************************************COMT       3
      INTEGER TMAX                                                      COMT       4
      PARAMETER (TMAX=35)                                               COMT       5
      COMMON /TYPEN/TDIM(TMAX,4),TTOT,TDB2(TMAX,2),TDBTOT,TERR          COMT       6
      COMMON /TYPEC/TYPE(TMAX,3),TDB1(TMAX)                             COMT       7
      INTEGER TTOT,TDBTOT,TERR                                          COMT       8
      REAL TDIM,TDB2                                                    COMT       9
      CHARACTER * 3 TYPE,TDB1                                           COMT      10
*=================================================                      COMT      11
* DESCRIPTION OF ARRAYS                                                 COMT      12
*=================================================                      COMT      13
*   ID        MATERIAL    FRAME MATERIAL                                COMT      14
*-------------------------------------------------                      COMT      15
*TYPE(X,1)   TYPE(X,2)   TYPE(X,3)                                      COMT      16
* A3          A3          A3                                            COMT      17
*=================================================                      COMT      18
* HEIGHT      WIDTH       LAYER       DISTANCE                          COMT      19
*                         THICKNESS   ABOVE FLOOR                       COMT      20
*-------------------------------------------------                      COMT      21
* TDIM(X,1)   TDIM(X,2)   TDIM(X,3)   TDIM(X,4)                         COMT      22
*  F8.2        F8.2        F8.2        F8.2                             COMT      23
*=================================================                      COMT      24
*  ID       ATTENUATION    AREA                                         COMT      25
```

```
* ----------------------------------------------                  COMT      26
* TDB1(X)     TDB2(X,1)    TDB2(X,2)                              COMT      27
*   A3           E9.3          E9.3                               COMT      28
**********************************************************************COMT  29
**********************************************************************COMT  30
      CHARACTER * 3, ID, MATID                                    RESOND    14
      DO 10 I = 1, TTOT                                           RESOND    15
      J=I                                                         RESOND    16
      IF(TYPE(I,1) .EQ. ID) GOTO 20                               RESOND    17
   10 CONTINUE                                                    RESOND    18
      IWARN = 11                                                  RESOND    19
      CALL WARNING(IWARN)                                         RESOND    20
      DREFLW = 0.0                                                RESOND    21
      RETURN                                                      RESOND    22
   20 CONTINUE                                                    RESOND    23
      MATID = TYPE (J,3)                                          RESOND    24
      RH = TDIM (J,1)                                             RESOND    25
      RW = TDIM (J,2)                                             RESOND    26
      REFL = RCOEF( MATID, FREQ, RFLAG)                           RESOND    27
      IF( REFL .LT. 0.80 ) THEN                                   RESOND    28
      DREFLW = 0.0                                                RESOND    29
      RETURN                                                      RESOND    30
      ENDIF                                                       RESOND    31
      A = RH                                                      RESOND    32
      B = RW                                                      RESOND    33
      IF( A.GT.B) THEN                                            RESOND    34
      TMP = B                                                     RESOND    35
      B = A                                                       RESOND    36
      A = TMP                                                     RESOND    37
      ENDIF                                                       RESOND    38
      FLOW =  3.0E8 / 2/ B                                        RESOND    39
      FHIGH = 3.0E8 / 2.0 * 3.0 * SQRT( 1/(B*B) + 1 / (A*A) )     RESOND    40
      IF(FREQ.GE.FLOW .AND. FREQ.LE.FHIGH) THEN                   RESOND    41
      DREFLW = 20.0                                               RESOND    42
      ELSE                                                        RESOND    43
      DREFLW = 0.0                                                RESOND    44
      ENDIF                                                       RESOND    45
      RETURN                                                      RESOND    46
      END                                                         RESOND    47
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

 A          205B                             REAL
 AFLAG        2B   /INITILN/                 REAL
 B          206B                             REAL
 BLDG         0B   /INITILC/                 CHAR*5
 DDABS        0B   /ROOMD/                   REAL          676
 DREFL     1244B   /ROOMD/                   REAL
 DREFLW    1245B   /ROOMD/                   REAL
 FERR        66B   /INITILN/                 INTEGER
 FHIGH      211B                             REAL
 FLOW       210B                             REAL
 FREQ         0B   /INITILN/                 REAL
 FREQA        4B   /INITILN/                 REAL           50
 FTOT        67B   /INITILN/                 INTEGER
 I          176B                             INTEGER
 ID           1    DUMMY-ARG                 CHAR*3
 IWARN      201B                             INTEGER
 J          200B                             INTEGER
 MATID      175B                             CHAR*3
 NROOMS    1244B   /ROOMN/                   INTEGER
 QUALITY      1B   /INITILN/                 INTEGER
```

```
 RAREA     1245B  /ROOMN/                    REAL          20
 REFL       204B                             REAL
 RFLAG        3B  /INITILN/                  REAL
 RH         202B                             REAL
 ROOM         0B  /ROOMN/                    REAL         676
 RW         203B                             REAL
 TDBTOT     323B  /TYPEN/                    INTEGER
 TDB1        37B  /TYPEC/                    CHAR*3        35
 TDB2       215B  /TYPEN/                    REAL          70
 TDIM         0B  /TYPEN/                    REAL         140
 TERR       324B  /TYPEN/                    INTEGER
 TMP        207B                             REAL
 TTOT       214B  /TYPEN/                    INTEGER
 TYPE         0B  /TYPEC/                    CHAR*3       105


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX     INTEGER                        50
 RMAX     INTEGER                        20
 TMAX     INTEGER                        35


--PROCEDURES--(LO=A)
 -NAME------TYPE---------ARGS------CLASS-----

 RCOEF     REAL            3      FUNCTION
 SQRT      GENERIC         1      INTRINSIC
 WARNING                   1      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM        74
    20     43B                    79


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 RESOND      5B     1


--STATISTICS--

 PROGRAM-UNIT LENGTH            214B =  140
 CM LABELLED COMMON LENGTH     3227B =  1687
 CM STORAGE USED              61000B =  25088
 COMPILE TIME                 0.120 SECONDS
```

```
      SUBROUTINE LFREQ                                                  LFREQ        1
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LFREQ        2
*[[[                                                                [[[LFREQ        3
*[[[  LOAD THE CONTENTS OF THE FILE 'BXXXXXF' INTO ARRAYS FREQA.        LFREQ        4
*[[[                                                                [[[LFREQ        5
*[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[LFREQ        6
************************************************************************LFREQ        7
************************************************************************COMF         1
*** COMMON FOR INITIAL PARAMETERS                                    ***COMF         2
************************************************************************COMF         3
      INTEGER FMAX                                                      COMF         4
      PARAMETER (FMAX = 50)                                             COMF         5
      COMMON /INITILN/ FREQ, QUALITY, AFLAG, RFLAG, FREQA(FMAX), FERR,  COMF         6
     $                  FTOT                                            COMF         7
      COMMON /INITILC/ BLDG                                             COMF         8
      CHARACTER * 5 BLDG                                                COMF         9
      REAL FREQ, AFLAG, RFLAG, FREQA                                    COMF        10
      INTEGER QUALITY, FERR, FTOT                                       COMF        11
************************************************************************COMF        12
************************************************************************COMF        13
**************************************************                      LFREQ        9
*   DECLARATION OF VARIABLES                                            LFREQ       10
**************************************************                      LFREQ       11
      INTEGER GETLEN, R, C                                              LFREQ       12
      CHARACTER * 7 NAME, PFN                                           LFREQ       13
**************************************************                      LFREQ       14
*                                                                       LFREQ       15
**************************************************                      LFREQ       16
      NAME = 'B'//BLDG(1:GETLEN(BLDG))//'F'                             LFREQ       17
      PFN = NAME (1:GETLEN(NAME))                                       LFREQ       18
      FERR = 0                     :G                                   LFREQ       19
      CALL PF ('GET',0,PFN(1ETLEN(PFN)),'RC',FERR)                    LFREQ       20
      IF ( FERR .EQ. 0 ) THEN                                           LFREQ       21
        OPEN (UNIT=3, FILE=PFN, FORM='FORMATTED',                       LFREQ       22
     $         STATUS='OLD', ACCESS='SEQUENTIAL')                       LFREQ       23
        FTOT = 0                                                        LFREQ       24
        DO 10 R = 1,FMAX                                                LFREQ       25
          READ (3,1000,END=20) FREQA(R)                                 LFREQ       26
 1000     FORMAT(E12.7)                                                 LFREQ       27
          FTOT = FTOT + 1                                               LFREQ       28
 10     CONTINUE                                                        LFREQ       29
 20     CONTINUE                                                        LFREQ       30
        CLOSE(3,STATUS='DELETE')                                        LFREQ       31
      ELSE IF ( FERR .EQ. 2 ) THEN                                      LFREQ       32
        CALL WARNING (11)                                               LFREQ       33
      ELSE                                                              LFREQ       34
        CALL WARNING (12)                                               LFREQ       35
      END IF                                                            LFREQ       36
      RETURN                                                            LFREQ       37
      END                                                               LFREQ       38
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS--BLOCK-----PROPERTIES-------TYPE---------SIZE

  AFLAG       2B  /INITILN/                   REAL
  BLDG        0B  /INITILC/                   CHAR*5
  C         NONE              UNUSED/*S*      INTEGER
  FERR       66B  /INITILN/                   INTEGER
  FREQ        0B  /INITILN/                   REAL
  FREQA       4B  /INITILN/                   REAL           50
  FTOT       67B  /INITILN/                   INTEGER
  NAME      210B                              CHAR*7
  PFN       211B                              CHAR*7
```

```
 QUALITY      1B   /INITILN/                   INTEGER
 R          207B                               INTEGER
 RFLAG        3B   /INITILN/                   REAL


--SYMBOLIC CONSTANTS--(LO=A)
 -NAME----TYPE------------------------VALUE

 FMAX     INTEGER                         50


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

 GETLEN     INTEGER        1      FUNCTION
 PF                        5      SUBROUTINE
 WARNING                   1      SUBROUTINE


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF

    10  INACTIVE   DO-TERM         41
    20      63B                    42
  1000     120B    FORMAT          39


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

 LFREQ       5B     0


--I/O UNITS--(LO=A)
 -NAME--- PROPERTIES-------------

 TAPE3   AUX/FMT/SEQ


--STATISTICS--

 PROGRAM-UNIT LENGTH             215B =  141
 CM LABELLED COMMON LENGTH        71B =  57
 CM STORAGE USED               61000B =  25088
 COMPILE TIME                  0.085 SECONDS
```

Appendix 9.8 Blank Forms for Data Taking.

WALLS DATA FORM

BUILDING I. D. NUMBER____________________ DATE____________________

NAME____________________

LINE #	DIRECTION	FROM	TO	HEIGHT(m)	WIDTH(m)	THICKNESS(cm)	MATERIAL	COMMENT

HOLES DATA FORM

BUILDING I.D. NUMBER________________________________ DATE______________________________

NAME____

LINE #	DIRECTION	FROM	TO	ID	COMMENT

TYPES DATA FORM

(for windows and doors)

BUILDING I.D. NUMBER________________________ DATE________________________

NAME________________________

LINE	ID	HEIGHT(m)	WIDTH(m)	DISTANCE ABOVE FLOOR	THICKNESS(cm)	LAYER MATERIAL	FRAME MATERIAL

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM.

BIBLIOGRAPHIC DATA SHEET (See instructions)

1. PUBLICATION OR REPORT NO.: NBSIR 84-3009

2. Performing Organ. Report No.:

3. Publication Date: September 1984

4. TITLE AND SUBTITLE

BUILDING PENETRATION PROJECT

5. AUTHOR(S)

J. C. Wyss, W. J. Anson, R. D. Orr

6. PERFORMING ORGANIZATION (*If joint or other than NBS, see instructions*)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (*Street, City, State, ZIP*)

Communications Electronics Engineering Installation Agency
Fort Huachuca, Arizona 08613

10. SUPPLEMENTARY NOTES

[ ] Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (*A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here*)

This report documents a computer program which calculates building attenuation of electromagnetic radiation over the frequency range 10 kHz - 10 GHz. Attenuation (in dB) is computed from building shape, dimensions, room layout, and the electrical properties of construction materials; no electromagnetic measurements are required. Details of the structure and use of the program are given.

12. KEY WORDS (*Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons*)

computer model; electromagnetic attenuation; electromagnetic interference; electromagnetic shielding; shielding materials

13. AVAILABILITY

[X] Unlimited
[ ] For Official Distribution. Do Not Release to NTIS
[ ] Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
[X] Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES: 310

15. Price: $25.00

USCOMM-DC 6043-P80