COMPUTER SIMULATION OF AIR-CONDITIONING SYSTEM DESIGN AND DUCTING ANALYSIS FOR PROFESSIONALS AND ENGINEERING STUDENTS

Isaac F. Odosola and Olufenti Ogunfolu

Abstract

Air-conditioning system design and ducting analysis has over the years been an aspect of Engineering, which even though has developed greatly, yet it brings about very tedious tasks of analysis, low level of accuracy, and a lot of time input in carrying out its analysis manually. This work makes provision for the use of a software package, designed for ducting analysis. It has the capability of handling analysis for buildings having as much as nine (9) floors, with each floor having up to twenty (20) rooms. The report gives an account of the method used and the programming aspect of the package for ducting analysis. The thermal analysis of a building forms the basis for the equipment selection in terms of capacity of the cooling equipment, quantity of dehumidified air required and the type of system to be recommended. The estimated air quantity is then used in carrying out the analysis of the system. The package has been designed in such a way that results of analysis made could be printed out for use. Also, it can be easily used on any computer that has a floppy drive for its installation, and has been produced using Visual Basic 6.0. it would be very useful for professionals, as it can carry out analysis which might take hours to develop in few seconds, and also, can be used in higher institutions as teaching aid to INSTRUCT Engineering students.

Introduction

Air-conditioning is the science and practice of creating a controlled climate in indoor space. It thus implies the simultaneous control of temperature, humidity, air movement and quality of air in an indoor space. In general, it includes any treatment of air to desired quality level.

It is worth noting that good air-conditioning cannot be achieved without proper duct design or analysis. The function of a duct system is to transmit air from the air handling apparatus to the space to be conditioned. To fulfill this function in a practical manner, the system must be designed within prescribed limits of available space, friction loss, velocity, sound level, heat and leakage losses and gains. Deficiencies in a duct system can result in systems that operate incorrectly, or are expensive to own and operate. Poor air distribution can cause discomfort, poorly designed sections of ductwork can result in unbalanced systems, and faulty duct construction produces inadequate air flow rates at the terminal units.

Presently, air-conditioning provided in many buildings is far from perfect. The limitations are not due generally to lack of knowledge, but to 'cutting corners' in design, either for economic reasons or the tedious task of carrying out proper analysis. The air-conditioning system designer is concerned about workable designs, and much research has been carried out on system performance and human comfort. Considerable emphasis is now being placed on design standards, which ultimately will be reflected in rigorous specifications, if not regulations, as noted by the American Society of Heating, Refrigeration and Air-conditioning Engineers (ASHRAE). This has ultimately led to the development of a software package, which takes advantage of the computer age to carry out ducting analysis for buildings, making use of a simple method and laid down specifications with minimal error in computation.

Duct Design Method

The general procedure for designing any duct system is to keep the layout as simple as possible. The supply terminals are located to provide proper room air distribution, and ducts are laid out to connect these outlets. The ductwork should be laid out or located to avoid structural members and equipment.

The design of a low velocity supply air system may be accomplished using any of the various methods, which include Velocity Reduction, Equal Friction and Static Regain methods. These various methods result in different levels of accuracy, economy and use. The Equal Friction method is used in the design of the package, due to its advantages and high level of accuracy. The usual procedure is to select an initial velocity in the main duct near the fan, with sound level being the limiting factor. The friction

chart is then used with the initial velocity and air quantity to determine the friction rate. This same friction loss is then maintained throughout the system, and the equivalent round duct diameter selected from the friction chart. In buildings where rectangular ducts are to be used, conversion tables in air- conditioning textbooks should be consulted in reading off values of the rectangular equivalent of round ducts. It should be noted that if rectangular duct sizes are determined directly from the duct area without using tables, the resulting duct sizes will be smaller, and velocity and friction loss will be greater for a given air quantity than the design values.

$$A = Q/V$$

or

$$Q = A.V$$

$$\text{But } A = \Pi\, d_e\, 2/4$$

$$\Rightarrow \quad d_e^2 = 4A/\Pi$$

$$\text{or} \quad d_e = 2\sqrt{A/\Pi}$$

where Q = total dehumidified air quantity (nrVhr)
V = duct velocity (m/hr)
A = duct-area (m^2) de =
duct diameter (m)

Design Of Software Package

The ducting analysis software was produced using Visual Basic 6.0. which is a programming language. The software has three (3) forms: one (I) for the introduction, and two (2) main forms for its operation. The first main form takes in the value of the total air quantity for a building for which ducting analysis is required and also the duct velocity, whose recommended value for various types of buildings can be got from air-conditioning textbooks. It normally ranges between 1200 and 2200 fpm for low velocity supply air systems. The duct area, and thus diameter of the main duct from the air handling unit/apparatus is thus calculated by the programme, using the equations above. After calculation of these values for the main duct, the user then proceeds to carry out the floor analysis. The package can adequately handle analysis for buildings having up to nine (9) floors at any particular time. Immediately the air quantity to each floor is fed in, the programme automatically calculates and shows the value of the duct diameter for each. After the floor analysis, the user could then print out the results, using 'Print Excel' command button and/or proceed to carry out the room analysis by clicking on the 'Room Analysis' command button. This takes the programme to the second form. Here, the user only inputs the floor number for which room analysis is required, and the values of that floor air quantity, its duct area and diameter are thus recalled. Thereafter, the user can proceed to the room analysis by feeding in the air quantity required in each room. As this is being done, the duct diameter is being calculated. The programme can handle room analysis for up to twenty (20) rooms at any particular time for a floor.

In generating results while using the package, values which relates the section area (%) to the air quantity (%) were used in the 'module' for the floor and room analysis. The S. I. Units is to be used when running the programme. Normally, the recommended ***minimum*** round duct diameter for ducting is 0.21m, or 210mm 8.4". Thus, calculated values which were less than this were automatically equated to 0.21m. Fig. 2 shows the flow chart for the software programme.

To validate the package, Kenneth Dike Library Extension, University of Ibadan was used as a case study. The building has three (3) floors, namely the basement, the lower ground floor and the upper ground floor. From the thermal analysis of the building, a dehumidified air quantity of 36,825nvVhr

(2l,675cfm) was estimated. A duct velocity of 3l,090m/hr (I700fpm) was selected from the range of recommended duct velocity for main ducts for libraries.

Results

Below is the analysis (obtained manually) for the Kenneth Dike Library Extension, using the section area (%) to the air quantity (%) relation rather than the friction chart:-

Floor		Air quantity
	basement	
1.	lower grouqd floor (LGF)	2,237nrVhr
2.	upper ground floor (UGF) to	18,729m^3/hr
3.	size the floor ducts	15,859nrVhr

Mains:

Total air quanti

Duct velocity,

Duct area,

Table 1.1

Duct diameter, d_e $= 2\sqrt{A/\Pi}$

$= 2\sqrt{1.185/3.142}$

$= 1.230m$

Floor:

Duct Section	**Air Otv(m^3/hr)**	**Cfm%**	**Duct Area %**	**Area (m'')**	**Duct Size (m)**
Mains	**36.825**	**100**	**100**	**1.185**	**1.23**
Basement	**2,237**	**6**	**10.5**	**0.124**	**0.40**
LGF	**18.729**	**51**	**59.0**	**0.699**	**0.94**
UGF	**15,859**	**43**	**51.0**	**0.604**	**0.88**

Room:

A. Basement floor has one (1) space or room (Bindery) Table 2.1

Duct Section	**Air Qty(m^3/hr)**	**Cfm%**	**Duct Area %**	**Area (m^2)**	**Duct Size (m)**
Basement (Bindery)	**2,237**	**100**	**100**	**0.124**	**0.40**

B. Lower ground floor (LGF) has fourteen (14) rooms.

Table 2.2

Duct Section	Air Qty(ni³/hr)	Cfm%	Duct Area %	Area (m²)	Duct Size (m)
I.OF	18.729	100	100	0.699	0.94
Document Librarian	58	0.3	*	0.035	0.21
Chief Cataloguer	44	0.2	*	0.035	0.21
General office	1.392	7.9	11.5	0.080	0.32
Mail office	89	0.5	2.0	0.014	0.21
Serial librarian	76	0.4	*	0.035	0.21
Senior Ass. Registrar	89	0.5	2.0	0.014	0.21
Accounts room	208	1.1	2.0	0.014	0.21
Deputy Lib. (tech. Services)	327	1.8	3.5	0.025	0.21
Sec. To dep. Librarian	199	1.1	2.0	0.014	0.71
Deputy librarian	100	0.5	2.0	0.014	0.21
Conference room	998	5.3	9.0	0.063	0.28
Microfilm room	348	1.9	3.5	0.025	0.21
Coffee room	1,048	5.6	10.5	0.073	0.31
Main reading Room	13,753	73.4	79.0	0.552	0.84

C. Upper ground floor has four (4) rooms **Table 2.3**

Duct Section	Air Oty(m³/hr)	Cfm%	Duct Area %	Area (in²)	Duct Size (m)
IJOF	15.859	100	100	0.604	0.88
Catalogue & Orders section	1,100	7	1 1.5	0.070	0.30
Entrance/ Reception area	1,347	9	14.5	0.088	0.34
Computer Appl. Room	803	5	9.0	0.054	0.26
Reading section	12,609	79	84.0	0.507	0.80

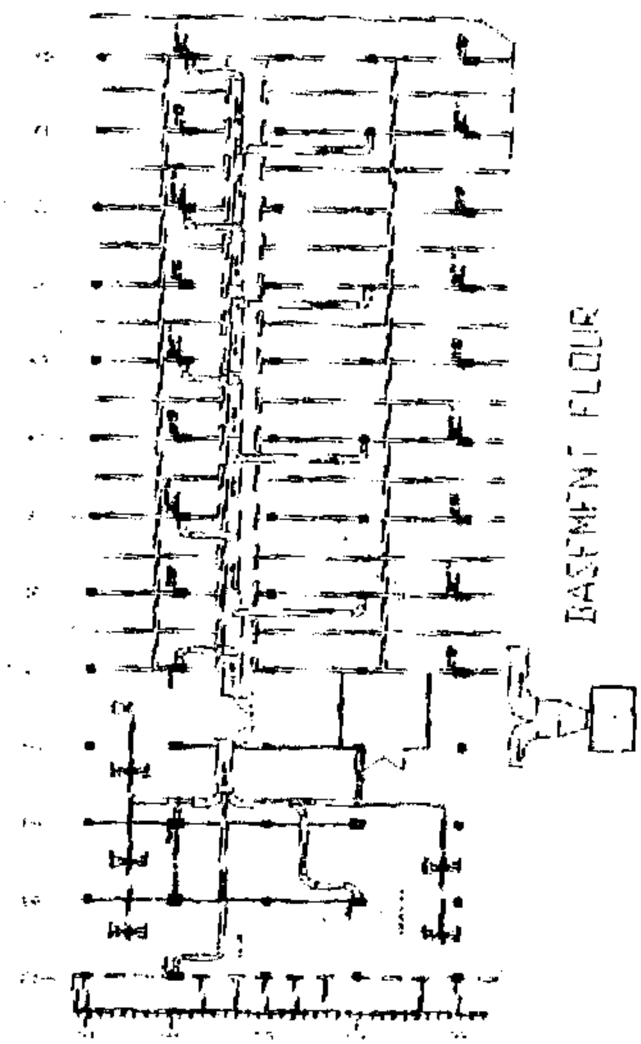
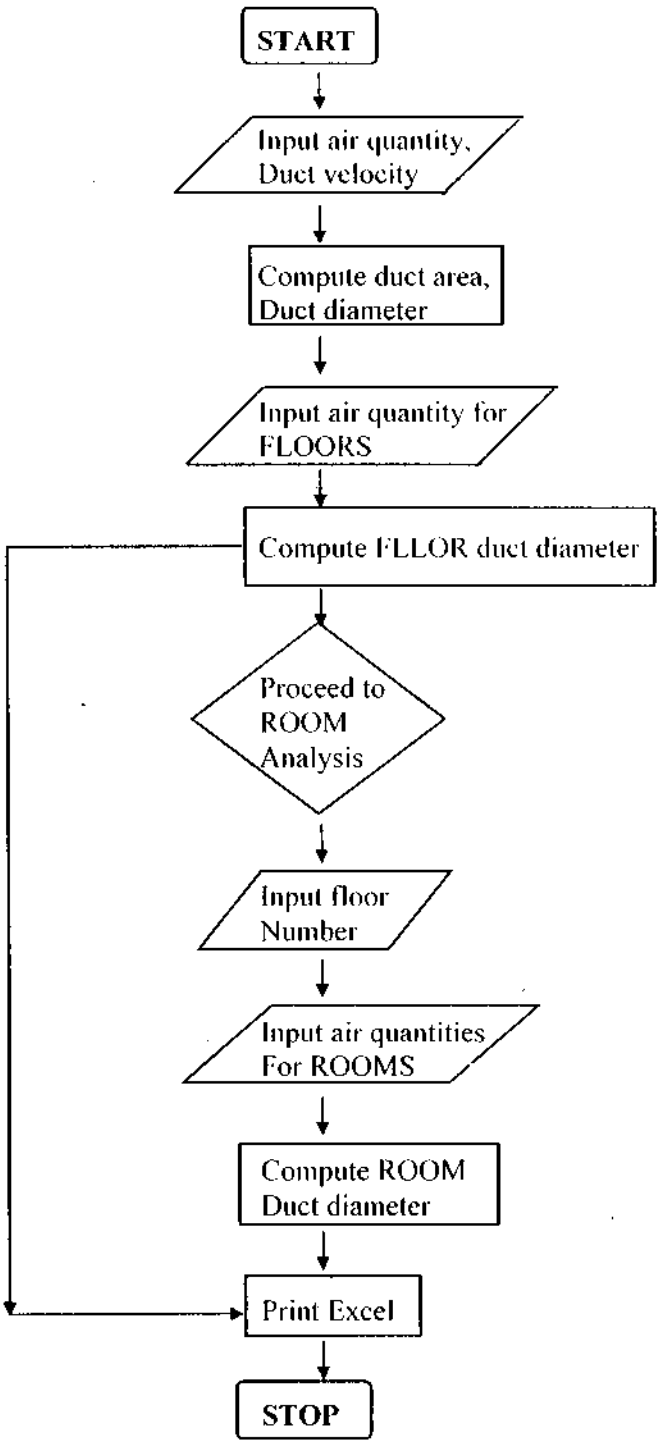


Fig. 1: FLOOR PLAN OF THE BASEMENT FLOOR SHOWING THE DUCT LAYOUT
Fig. 2 FLOW CHART: Ducting Analysis



Discussion

Tables 1.1. 2.1, 2.2 and 2.3 show the results of the hand-calculated analysis for the three floors from the mains and the room analysis for the Basement, Lower ground floor and Upper ground iloor respectively. These, when compared with the results of the computer print-out of results from the software package analysis showed that the same values were got, i.e. both methods were accurate and realistic. Obviously, the hand-calculated analysis is more rigorous and time consuming when compared with the computer simulation, which was processed in seconds. Duct area % less than 2%, represented as

* in the hand-calculations of table 2.2, automatically gave the least duct diameter of 0.21m in the software. The floor plan layout of the Basement floor is shown in fig. I, while fig. 2 shows the flowchart for the programme.

Conclusion

The efficient and proper functioning of an air-conditioning system depends to an extent on the duct design and analysis. Most often, sound level, such as in a library, has to be the limiting factor. Noise generated in ducts depend greatly on the duct velocity used in a system. Thus, the effectiveness of the software depends on proper selection and/or calculations of the basic design parameters. The software itself has proven to be useful and effective in terms of accuracy, time factor and tedious task of hand- calculations for analysis. Thus it becomes a ready tool for professionals and engineering students of higher institution.

Appendix 1

Source Code For Ducting Analysis (Visual Basic 6.0)

```
option Explicit

Private Sub Form_KeyPress(Key Ascii As Integer)
'this function starts the programme on pressing
'any key Unload Me Form 1 .Show- End Sub

Private Sub Framel_Click()
'Unload Me Paid Sub

Private Sub TextlLostFocusQ ' DAQ represents the
total dehumidified air quantity On Error Resume
Next DAQ = CDbl (Text 1.Text)
End Sub

Private Sub Text2_Change()
'vel represents the velocity, area I the area of main
duct VEL = CDb!(Test2.Text)
AREA I = DAQ/VEL Text3.Text = Round(AREA
1,3)
DIA1 =(4* AREA 1 (3,142)" 0.5 Text4.Text =
Round(DIAI, 3)
End Sub
' DAQ I,DAQ2,... represents the air quantity
'required on each floor, and D1AF1,D1AF2,...
'the diameter of the branch ducts to the floors
private Sub Text5_LostFocus()
On Error Go To FEM DAQ I = CDbl(Text5.Te.xt)
a - CFM(DAQI)
CFMI = CFMF B = DUCT(CFMI)
AREAFI = AREAF AREAF = 0
DIAF1 = Sqr(4 * AREAFI) / 3.142) lf(DIAF I <0.21) Then Textl4.Text = 0.21

AREAF2 = 0 D1AF2 = 0 End Sub
Private Sub Textl7_LostFocus()
On Error Go To FEM DAQ3 =
CDbl(Text7.Text) a =
CFM(DAQ3)
CFM3-CFMF B = DUCT(CFM3)
AREAF3 = AREAF AREAF=0
D1AF3 = Sqr((4 * AREAF3) /
3.142) If (DIAF3 <0.21) Then
Textl7.Text = 0.21 Else
Textl7.Text = Round(DlAF3, 3)
End If Exit Sub FEM:
AREAF3 =0 DIAF3 = 0 End Sub
Private Sub Tex8_LostFocus()
On Error Go To FEM a =
CFM(DAQ4)
CFM4 = CFMF B =
DUCT(CFM4)
AREAF4 = AREAF AREAF=0
DIAF4 = Sqr((4 * AREAF4) /
3.142) If (DIAF4 <0.21) Then
Text 18.Text = 0.21 Else
Text 18.Text = Round(DIAF4, 3)
End If Exit Sub FEM:
AREAF4 = 0 DIAF4 = 0 End Sub
```

```
Else
Text.14.Text = Round(DIAFl, 3)
End If Exit Sub FEM:
AREAF1 =0 End Sub
Private Sub Text6_LostFocus()
On Error Go To FEM DAQ2 =
CDbl(Text6.Text) a =
CFM(DAQ2)
CFM2 = CFMF B =
DUCT(CFM2)
AREAF2 = AREAF AREAF = 0
DIAF2 = Sqr((4 * AREAF2) /
3.142) If(DIAF2 <0.21) Then
Textl6.Text = 0.21 Else
Text 16.Text = Round(DIAF2, 3)
End If Exit Sub FEM:
B = DUCT(CFM6)
AREAF6 = AREAF AREAF=0
DIAF6 = Sqr((4 * AREAF6)./
3.142) If (DIAF6 <0.21) Then
Text20.Text = 0.21 Else
Text20.Text = Round(DIAF6, 3)
End If Exit Sub FEM:
AREAF6 = 0 DIAF6 = 0 End Sub
Private Sub Text! l_LostFocus()
On Error Go To FEM DIAQ7 =
CDbl(TextI l.Text) a =
CFM(DAQ7)
CFM7 = CFMF B =
DUCT(CFM7)
AREAF7 = AREAF AREAF = 0
DIAF7 = Sqr((4 * AREAF7) /
3.142) If (DIAF7 <0.21) Then
Text21.Text = 0.21 Else
Text21 .Text = Round(DIAF7, 3)
End If Exit Sub FEM:
AREAF7 = 0 DIAF7 = 0 End Sub
Private Sub Textl2_LostFocus()
On Error Go To FEM
Private Sub Text9_LostFocus()
On Error Go To FEM DAQ5 =
CDb 1 (Text9 .Text) a =
CFM(DAQ5)
CFM5 = CFMF B =
DUCT(CFM5)
AREAF5 = AFEA AREAF = 0
DIAF5 = Sqr((4 * AREAF5) /
3.142) If (DIAF5 <0.21) Then
Textl9.Text = 0.21 Textl9.Text =
Round (DIAF5, 3)
End If Exit Sub FEM:
AREAF5 = 0 DIAF5 = 0 End Sub
Private Sub TextlOLostFocusf)
On Error Go To FEM DAQ6 =
CDb 1 (Text lO.Text) a =
CFM(DAQ6)
CFM6 = CFMF DIAF9 = 0 End
Sub
corrwks.cells(l, 1). Value =
corrwks.cells(l, 2). Value =
corrwks.ceIls(l, 3). Value =
corrwks.cells(l, 4). Value =
corrwks.cells(2, 1). Value =
corrwks.cells(2, 2). Value =
corrwks.cells(2, 3). Value =
corrwks.cel!s(2,4). Value =

Private Sub Command I _Click()
'this function takes the programme to the next form for room
analysis FRMFLOOR.Show End Sub

Private Sub Command3_Click()
'this function prints the results as Microsoft Excel
output Dim correxcel As Object Dim corrwks As
Object
Set correxcel = CreateObject("excel.application")
correxcel. workbooks. Add
Set corrwks = correxcel.activesheet
corrwks.cells(5, 1).              "Total air
Value = corrwks.cells(5,          qty(m^3/hr)" "Duct
2). Value =                       velocity(m/hr)"
corrwks.cells(5, 3).              "Duct area(m^2)"
Value = corrwks.cells(6,          "Dia of main
I). Value =                       duct(m)"
corrwks.cells(6, 2).              DAQ
Value = corrwks.cells(6,          VEL
3). Value =                       AREA1
corrwks.cells(7, 1).              DIA1
Value = corrwks.ceils(7,          "FLOOR NOS"
2). Value =                       "AIR QTY(m^3/hr)"
corrwks.cells(7, 3).              "DUCT DIA(m)"
Value =
                                  DAQ1
                                  DIAF1
                                  2
                                  DAQ2
                                  D1AF2
```

```
DAQ8 = CDbl(Textl2.Text) a =
CFM(DAQ8)
CFM8 = CFMF B =
DUCT(CFM8)
AREAF8 = AREAF AREAF = 0
DIAF8 = Sqr((4 * AREAF8) /
3.142) If (DIAF8 <0.21) Then
Text22.Text = 0.21 Else
Text22.Text = Round(DIAF8, 3)
End If Exit Sub FEM:
AREAF8 = 0 D1AF8 = 0 End Sub
Private Sub Textl3_LostFocus()
On Error Go To FEM DAQ9 =
CDbl (Text 13.Text) a =
CFM(DAQ9)
CFM9 = CFMF B =
DUCT(CFM9)
AREAF9 = AREAF AREAF=0
D1AF9 = Sqr((4 * AREAF9) /
3.142) If (D1AF9 <0.21) Then
Text23.Text = 0.21 Else
Text23.Text = Round(DIAF9,3)
End If Exit Sub FEM:
AREAF9 = 0 Elself (NOS = 2)
Then Text2.Text = DAQ2
Text3.Text = D1AF2 a = change()
Elself (NOS = 3) Then Text2.Text
= DAQ2 a = change()
Elself (NOS = 4) Then Text2.Text
= DAQ4 Text3.Text = DIAF4 a =
change()
Elself (NOS = 5) Then Text2.Text
= DAQ5 •
Text3.Text = DIAF5 a = change()
Elself (NOS = 6) Then Text2.Text
= DAQ6 Text3.Text = DIAF6 a =
change()
Elself (NOS = 7) Then Text2.Text
= DAQ7 Text3.Text = D1AF7
corrwks.cells(8, I). Value = “3"
ccrrwks.cells(8, 2). Value = DAQ3
corrwks.cells(8, 3). Value = D1AF3
corrwks.cells(9, I). Value = “4”
corrwks.cells(9, 2). Value = DAQ4
corrwks.cells(9, 3). Value = DIAF4
corrwks.cells(10, 1). Value = “5”
corrwks.cells(10, 2). Value = DAQ5
corrwks.cel!s(10, 3). Value = D1AF5
corrwks.cel!s(l 1, 1). Value = “6”
corrwks.cells(l 1,2). Value = DAQ6
corrwks.cells(l 1, 3). Value = DIAF6
corrwks.cells(12, 1). Value = “7”
corrwks.cells(12, 2). Value = DAQ7
corrwks.cells(12, 3). Value = D1AF7
corrwks.cells(13, 1). Value = “8”
corrwks.cells(13, 2). Value = DAQ8
corrwks.cells(13, 3). Value = DIAF8
corrwks.cells(14, I). Value = “9”
corrwks.cel!s(14, 2). Value = DAQ9
corrwks.cells(l4, 3). Value = DIAF9
correxcel.save correxcel.quit End Sub

Private Sub Text 1 Lost Focus()
‘this function takes the floor number that the
user ‘inputs, and automatically calls the air
quantity ‘and diameter of the duct for that
floor number NOS = CDbl (Text 1.Text)
If (NOS = 1) Then Text2.Text = DAQ1
Text3.Text = DIAF1 a = changeQ FEM:
AREAR11 =0 DIAR11 =0
End Sub FEM:
AREAR11 =0 End Sub
a = CFN(DAQR12)
CFMR12 = CFMF B = DUCT(CFMR12)
AREAR12 = AREAF AREAR = 0
D1AR12 = Sqr((4 * AREAR 12) / 3.142)
DIAR12 = confirm(DlAR12)
AREAR12 = AREAF
Exit Sub
FEM:
AREAR12 = 0 AREAR12 = AREAF End
Sub
Private Sub Text 18_LostFocus()
On Error Go To FEM
```

```
a = changeQ Elself (NOS = 8)
Then Text2.Text = DAQ8
Text3.Text = DIAF8 a =
change()
Elself (NOS = 9) Then
Text2.Text = DAQ9
Text3.Text = DIAF9 a =
change()
End If End Sub
'this function does the room analysis for
the 'floors, using the air quantity for
each room 'to determine the duct sizes
Private Sub Text 14_LostFocus()
On Error Go To FEM DAQR10 =
CDbl(TextM.Text) a = CFM(DAQRIO)
CFMR10 = CFMF B =
DUCT(CFMRIO)
AREAIO = AREAF AREAF = 0
DIAR10 = Sqr((4 * AREAR10) / 3.142)
DIAR10 = confirm(DIARlO)
Text35.Text = Round(DIARlO, 3)
Exit Sub FEM:
AREARI =0 DIAR1 =0 End Sub
Private Sub Textl6_LostFocus()
On Error Go To FEM DAQR11 =
CDbl(Textl6.Text) a = CFM(DAQR11)
CFMR11 = CFMF B = DUCT(CFMRI
1)
AREARI I = AREAF AREAF = 0
DIARII = Sqr((4* AREARI 1)/3.142)
DIAR11 = confirm(DIARl 1)
Text36.Text = Round(DIARl 1, 3)
Exit Sub FEM:
AREARI 5 = 0 DIARI5 = 0 End Sub
Private Sub Text21_LostFocus()
On Error Go To FEM DAQR16 = CDbl
(Text21 .Text) a = CFM(DAQR16)
CFMR16 = CFMF B =
DUCT(CFMR16)
AREARI 6 = AREAF AREAF = 0
DIAR16 = Sqr((4 * AREARI6) / 3.142)
D1AR16 = confirm(DIAR16)
AREARI2 = AREAF a =
CFM(DAQRI3)
CFMRI3 = CFMF B =
DUCT(CFMR13)
AREAR13 = AREAF AREAF =
0
DIAR13 = Sqr((4 * AREAR12) / 3.142)
DIARI3 = contirm(DIARI 3)
Text38.Text = Round(DlARI3, 3)
Exit Sub FEM:
AREAR13 = 0 DIAR13 = 0 End Sub
Private Sub Textl9_LostFocus()
On Error Go To FEM DAQR 14 = CDbl
(Text 19.Text) a = CFM(DAQR14)
CFMR14 = CFMF B =
DUCT(CFMR14)
AREAR14 = AREAF AREAF = 0
DIAR14 = Sqr((4 * AREARI4) / 3.142)
DIAR14 = confirm(DIAR14)
Text39.Text = Round(DIARI4, 3)
Exit Sub FEM:
AREAR 14 = 0 D1AR14 = 0 End Sub
Private Sub Text20_LostFocus()
On Error Go To FEM DAQR 15 = C Db
1 (T ext20.T ext) a = CFM(DAQR15)
CFMR15 = CFMF B =
DUCT(CFMR15)
AREARI 5 = AREAF AREAF = 0
DIAR15 = Sqr((4 * AREARI 5) / 3.142)
DIAR15 = confirm(DlAR15)
Text40.Text = Round(DlAR15, 3)
Exit Sub FEM:
AREAR 19 = 0 AREARI9 = 0 End Sub
Private Sub Text25_LostFocus()
On Error Go To FEM DAQR20 =
CDbl(Text25.Text) a = CFM(DAQR20)
CFMR20 = CFMF B =
DUCT(CFMR20)
AREAR20 = AREAF AREAF = 0
DIAR16 = Sqr((4 * AREAR20) / 3.142)
D1AR2G = confirm(DlAR20)
Tex.t41.Text = Round(DIAR16, 3)
Exit Sub FEM:
AREAR16 = 0 DIAR16 = 0 End Sub
Private Sub Text22_LostFocus()
On Error Go To FEM DAQRI7 = CDb 1
(T ext22.T ext) a = CFM(DAQR17)
CFMR17 = CFMF B = DUCT(CFMR
17)
AREAR17 = AREAF AREAF = 0
DIAR11= Sqr((4 * AREAR17) / 3.142)
DIAR17 = confirm(DIARI7) ,
Text4I.Text = Round(DIARl7, 3)
Exit Sub FEM:
AREAR17 = 0 DIAR17 = 0 End Sub
Private Sub Text23_LostFocus()
On Error Go To FEM DAQR18 =
CDbl(Text23.Text) a = CFM(DAQR18)
CFMRI 7 = CFMF B =
DUCT(CFMR18)
AREAR18 = AREAF AREAF=0
DIARI8 = Sqr((4 * AREAR18) / 3.142)
D1AR18 = confirm(DIAR18)
Text43.Text = Round(DlARI8, 3)
Exit Sub FEM:
AREARI8 = 0 D1ARI8 = 0 End Sub
Private Sub Text24_LostFocus()
On Error Go To FEM
DAQR19 = CDb 1 (Text24.Text)
a = CFM(DAQR19)
CFMR19 = CFMF B =
DUCT(CFMR19)
```

```
AREAR19 = AREAF AREAF=0
DIARI9 = Sqr((4 * AREAR19) / 3.142)
DIAR19 = confirirt(DIAR19)
Text44.Text = Round(DIAR19, 3)
Exit Sub FEM:


End Sub
Private Sub Text8_LostFocus() On Error
Go To FEM
Text45.Text = Round(DlAR20, 3)
Exit Sub
FEM:
AREAR20 = 0 D1AR20 = 0 End Sub
Private Sub Text22_LostFocus()
On Error Go To FEM DAQR1 = CDb
l(Text5.Text) a = CFM(DAQRl)
CFMRI = CFMF B = DUCT(CFMRI)
AREAR1 =AREAF AREAF=0
DIAR 1 = Sqr((4 * AREAR16) / 3.142)
DIAR1 = confirm(DIARl) Text26.Text
= Round(DlARl, 3)
Exit Sub FEM:
AREAR 1 =0 DIARI =0 End Sub
Private Sub Text6_LostFocus()
On Error Go To FEM DAQR2 =
CDbl(Text6.Text) a = CFM(DAQR2)
CFMRI = CFMF B = DUCT(CFMR2)
AREAR2 = AREAF AREAF = 0
DIAR2= Sqr((4 * AREAR2) / 3.142)
D1AR2 = confirm(DIAR2) Text27.Text
= Round(DIAR2, 3)
Exit Sub FEM:
AREAR2 = 0 DIAR2 = 0 End Sub
Private Sub Text7_LostFocus()
On Error Go To FEM DAQR3 =
CDbl(Text7.Text) a = CFM(DAQR3)
CFMR3 = CFMF B = DUCT(CFMR3)
AREAR3 = AREAF AREAF=0
D1AR3= Sqr((4 * AREAR3) / 3.142)
DIAR3 = confirm(DIAR3) Text28.Text
= Round(DlAR3, 3)
Exit Sub FEM:
AREA31 =0 DIAR3 =0
DAQR8 = CDbl(Textl2.Text) a=
CFM(DAQR8)
CFMR8 = CFMF
```

```
DAQR4 = CDbl(Text8.Text) a =
CFM(DAQR4)
CFMR4 = CFMF B =
DUCT(CFMR4)
AREAR4 = AREAF AREAF = 0
DIAR4 = Sqr((4 * AREAR4) /
3.142) DIAR4 = confirm(DIAR4)
Text29.Text = Round(DIAR4, 3)
Exit Sub FEM:
AREAR4 = 0 DIAR4 = 0 End Sub
Private Sub Textl9_LostFocus()
On Error Go To FEM DAQR5 =
CDbl(Text9.Text) a =
CFM(DAQR5)
CFMR5 = CFMF B =
DUCT(CFMR5)
AREAR5 = AREAF AREAF = 0
DIAR5 = Sqr((4 * AREAR5)
/3.142) D1AR5 = confirm(DlAR5)
Text30.Text = Round(DIAR5; 3)
Exit Sub FEM:
AREAR5 = 0 D1AR5 = 0 D1AR5
= 0 End Sub
Private Sub TextlO_LostFocus()
On Error Go To FEM DAQR6 =
CDbl(TextlO.Text) a =
CFM(DAQR6)
CFMR6 = CFMF B =
DUCT(CFMR6)
AREAR6 = AREAF AREARF = 0
DIAR6 = Sqr((4 * AREAR6)
/3.142) DIAR6 = confirm(DlAR6)
Text31.Text = Round(DIAR6, 3)
Exit Sub FEM:
AREAR6 = 0 DIAR6 = 0 End Sub
Private Sub Textl l_LostFocus()
On Error Go To FEM DAQR7 =
CDbl(Textl I.Text) a =
CFM(DAQR7)
CFMR7 = CFMF B =
DUCT(CFMR7)
AREAR7 = AREAF AREARF = 0
DIAR7 = Sqr((4 * AREAR7)
/3.142)
```

```
B = DUCT(CFMR8)
AREAR8 = AREAF AREAF
= 0
DIAR8 = sqr((4 * AREAR8) /
3.142) DIAR8 = confirm(DIAR8)
Text33.Text = Round(DIAR8, 3)
Exit Sub FEM:
AREAR8 = 0 D1AR8 = 0 End Sub
Private Sub Textl3_LostFocus()
On Error Go To FEM DAQR9 =
CDbl (Text 13.Text) a =
CFM(DAQR9)
CFMR9 = CFMF B =
DUCT(CFMR9)
AREAR9 = AREAF AREAF=0
DIAR9 = Sqr((4 * AREAR9) /
3.142) D1AR9 = confirm(DlAR9)
Text34.Text = Round(DIAR9, 3)
Exit Sub FEM:
AREAR9 = 0 DIAR9 = 0 End Sub
corrwks.cells(l,    1).    Value
corrwks.cells(l,    2).    Value
corrwks.cells(  1,   3).   Value
corrwks.cells(2,    1).    Value
corrwks.cells(2,    2).    Value,
corrwks.cells(2,    3).    Value
corrwks.cells(5,    1).    Value
corrwks.cells(5,    2).    Value
```

```
Private Sub Commandl_Click()
'this function prints the results as Microsoft Excel
output Dim correxcel As Object Dim corrwks As
Object
Set correxcel = CreateObject("excel.application")
correxcel. workbooks. Add
Set corrwks = correxcel.activesheet
```

```
corrwks.celIs(5, 3). Value corrwks.cells(6, 1). Value
corrwks.celis(6, 2). Value corrwks.cells(6, 3). Value
corrwks.cells(7, 1). Value corrwks.cells(7, 2). Value
corrwks.cells(7, 3). Value corrwks.cells(8, I). Value
corrwks.cells(8, 2). Value corrwks.cells(8, 3). Value
corrwks.cells(9, I). Value corrwks.cells(9, 2). Value
```

```
"Floor air
qty(m^3/hr)" "Total
nos"
"Duct diameter(m)"
Text2.Text "Textl
.Text "Text3.Text
"ROOM NOS"
"AIR QTY(m^3/hr)"
"DUCT DIA(m)"

Text5.Text
Text26.Text

Text6.Text
Text27.Text
"3"
Text7.Text
Text28.Text
"4"
Text8.Text
```

```
DIAR7 = conflrm(DIAR7)
Text32.Text = Round(DIAR7, 3)
Exit Sub FEM:
AREAR7 = 0
D1AR7 = 0 End
Sub
Private Sub Textl2_LostFocus()
On Error Go To FEM
corrwks.ce  eorrwks.ee  corrwks.ee
corrwks.c   s( 12,3). Value =
e           Text32.Text  1). Value = "8"
corrwks.ce      2). Value = Textl 2.Text
corrwks.c       3). Value = Text33.Text
e           s(  1). Value = "9"
corrwks.c   14, 2). Value = Textl 3.Text
e               3). Value = Text34.Text
corrwks.c   s(  1), Value = "10"
e           15, 2). Value = Textl4.Text
corrwks.ce      3). Value = Text35.Text
corrwks.c   s(  1). Value = "11"
e           16, 2). Value = Textl6.Text
corrwks.c   s(  3). Value = Text36.Text
e           16, 1). Value = "12"
corrwks.c   s(  2). Value = Textl7.Text
e           16, 3). Value = Text37.Text
corrwks.c   s(  1). Value = "13"
e           17, 2). Value = Textl8.Text
corrwks.c   s(  3). Value = Text38.Text
e           17, ))• Value = "14"
corrwks.c   s(l 2). Value = Textl9.Text
e           7,  s( 3). Value = Text39.Text
corrwks.c   18, 1). Value = "15"
e           s(20 2). Value = Text20.Text
corrwks.ce  .   3). Value = Text40.Text
corrwks.ce      1). Value = "16"
corrwks.ce      2). Value = Text21 .Text
corrwks.ce      3). Value = Text41 .Text
corrwks.c   s(22 1). Value = "17"
e           ,   2). Value = Text22.Text
corrwks.c   s(22 3). Value = Text42.Text
e           ,   1). Value = "18"
corrwks.c   s(22 2). Value = Text23.Text
e           ,   3). Value = Text43.Text
corrwks.c   s(23 !) Value = "19"
e           ,   2). Value = Text24.Text
corrwks.c   s(23 3). Value = Text44.Text
e           s(25 !)• Value = "20"
corrwks.c   s(23 2). Value = Text25.Text
e           ,   3). Value = Text45.Text
corrwks.c
e     corrwks.ce
corrwks.ce
corrwks.ce  corrwks.ce  corrwks.ce
corrwks.ce  corrwks.ce  corrwks.ce
corrwks.ce  corrwks.ce  corrwks.ce
corrwks.ce              corrwks.ce
correxcel.save  correxcel.quit  End

corrwks.cells(9, 3). Value =
"Text29.Text corrwks.cells(IO, I).
Value = "5" corrwks.cells(IO, 2). Value
= Text9.Text corrwks.cells(10, 3).
Value = Text30.Text corrwks.cells(l
1,1). Value = Text6.Text corrwks.cells(l
I, 2). Value = TextIO.Text
corrwks.cells(l 1, 3). Value =
Text31.Text corrwks.cells(12, 1). Value
= Text7.Text corrwks.cells(12, 2).
Elself (S = 7) Then        REAF = DA *
  DA = (11.5/100)
AREAF = DA * AREAI   Sub
Elself (S = 8) Then
  DA = (13/ 100)
AREAF = DA * AREAI
Elself (S = 9) Then
  DA = (16.5/ 100)
AREAF = DA * AREAI
Elself (S = 10) Then
  DA = (16.5 / 100)
AREAF= DA * AREAI
Elself (S = 11) Then
  DA = (17.5/ 100)
AREAF = DA *AREAI
Elself (S= 12) Then
  DA = (18.5/ 100)
AREAR = DA * AREAI
Elself (S = 13) Then
  DA = (19.5/ 100)
AREAF = DA * AREAI
Else (S = 14) Then
  DA (20.5/100)
AREAR = DA * AREAI
Elself (S = 15) Then
  DA = (21.5/ 100)
AREAF = DA * AREAI
Elself (S = 16) Then
  DA = (23 / 100)
AREAF= DA * AREAI
Elself (S = 17) Then
  DA = (24/ 100)
AREAF = DA * AREAI
Elself (S = 18) Then
  DA = (25 / 100)
AREAF = DA* AREAI
Elself (S = 19) Then
  DA = (26/100)
AREAF = DA *AREAI
```

ElseIf(S = 20)
Then DA =
(27/100)
AREAF = DA * AREAI ElseIf(S =
21) Then DA = (28 / 100)

```
Public DAQ, VEL, AREA I, DIA1, CFMF, AREAF,
DIAF1, DIAF2, DIAF3, DIAF4, DIAF5, DIAF6,
DIAF7, D1AF8, DIAF9, DAQ1, DAQ2, DAQ3, DAQ4,
DAQ5, DAQ6, DAQ7, DAQ8, DAQ9, As Double
Public Function CFM(W)
CFMF = Round((W * 100 / DAQ), 0)
End Function
Public Function DUCT(S)
If (S = I) Then DA = (2/ 100)
AREAF = DA * AREA I Elself (S = 2) Then DA =
(3.5/100)
AREAF = DA * AREA 1 Elself (S = 3) Then DA =
(5.5/100)
AREAF = DA * AREAI Elself (S = 4) Then DA = (7 /
100)
AREAF = DA * AREAI Elself (S = 5) Then DA =
(9/100)
AREAF = DA *AREAI Elself (S = 29) Then DA =
(36.5 /100)
Elself (S = 2) Then DA = (3.5/ 100)
AREAF= DA *AREAI Elself (S = 30) Then DA = (37.5
/ 100)
AREAF = DA * AREAI Elself (S = 31) Then DA = (39/
100)
AREAF = DA * AREAI Elself (S = 32) Then DA =
(40/100)
AREAF = DA * AREAI Elself (S = 32) Then DA = (40/
100)
AREAF = DA* AREAI Elself (S = 33) Then DA = (41
/100)
AREAF = DA * AREAI Elself (S = 34) Then DA = (42/
100)
AREAF= DA * AREAI Elself (S = 35) Then DA = (43
/100)
AREAF = DA * AREAI Elself (S = 36) Then DA = (44/
100)
AREAF= DA *AREAI Elself (S = 37) Then DA = (45/
100)
AREAF = DA * AREAI Elself (S = 38) Then
```

```
AREAF = DA *
AREAI ElseIf(S = 22)
Then DA = (29.5 /
100) AREAF = DA *
AREAI ElseIf(S = 23)
Then DA = (30.5 /
100) AREAF = DA *
AREAI EIseIf(S= 24)
Then DA = (31.5/ 100)
AREAF = DA *
AREAI ElseIf(S= 25)
Then DA = (32.5 /
100) AREAF = DA *
AREAI ElseIf(S= 26)
Then DA = (33.5 /
100) AREAF= DA *
AREAI ElselffS= 25)
Then DA = (34.5 I
100) AREAF = DA *
AREAI ElseIf(S= 28)
Then DA = (35.5 /
100) AREAF = DA *
AREA!
```

```
DA = (59 / 100)
AREAF = DA *
AREAI Elself(S= 52)
Then DA = (60/100)
AREAF = DA *
AREAI ElseIf(S= 53)
Then DA = (61 / 100)
AREAF = DA *
AREAI Elself(S= 54)
Then DA = (32.5 /
100) AREAF = DA *
AREAI Elself (S= 62)
Then DA = (32.5 /
100) AREAF= DA *
AREAI Elself (S= 55)
Then DA = (63 / 100)
AREAF = DA *
AREAI Elself (S=56)
Then DA = (64/ 100)
AREAF = DA *
AREA! Elself (S= 57)
Then DA = (65 / 100)
AREAF = DA *
AREAI Elself (S= 58)
Then DA = (65.5 /
100) AREAF = DA *
AREAI Elself (S= 59)
Then DA = (66.5/ 100)
AREAF = DA *
AREAI Elself (S= 60)
Then DA = (67.5 /
100)
```

DA = (46/ 100)
AREAF = DA *
AREA1 Elself (S =
39) Then DA = (47/
100)
AREAF = DA
*AREA1 Elself (S =
40) Then DA = (48/
100)
AREAF = DA *
AREA! Elself (S = 41)
Then DA = (49 / 100)
AREAF= DA *
AREAI Elself (S = 42)
Then DA = (50/ 100)
AREAF = DA *
AREAI Elself (S = 43)
Then DA = (51/ 100)
AREAF = DA *
AREAI Elself (S = 44)
Then DA = (52 / 100)
AREAF = DA *
AREAI Elself (S = 45)
Then DA = (53 / 100)
AREAF = DA *
AREAI Elself (S = 46)
Then DA = (54 / 100)
AREAF = DA *
AREAI Elself (S = 47)
Then DA = (55 / 100)
AREAF = DA *
AREAI Elself (S = 48)
Then DA = (56/ 100)
AREAF = DA *
AREAI Elself (S = 49)
Then DA = (57 / 100)
AREAF = DA *
AREAI Elself (S = 50)
Then DA = (58 / 100)
AREAF = DA*
AREAI Elself (S = 51)
Then DA = (79/ 100)
AREAF = DA *
AREAI Elself (S= 74)
Then DA = (80/ 100)
AREAF = DA *
AREAI Elself (S= 75)
Then DA = (80.5 /
100) AREAF = DA *
AREAI Elself (S= 76)
Then DA = (81 / 100)
AREAF = DA *
AREAI Elself (S= 77)
Then DA = (82 / 100)
AREAF = DA *
AREAI Elself (S= 78)

```
Then DA = (83 / 100)
AREAF = DA * AREAI
```

```
AREAF = DA*
AREAI Elself (S= 61)
Then DA = (68/ 100)
AREAF = DA *
AREAI Elself (S= 62)
Then DA = (69/ 100)
AREAF = DA *
AREAI Elself (S= 63)
Then DA = (70/ 100)
AREAF= DA *
AREAI Elself (S= 64)
Then DA = (71 / 100)
AREAF = DA *
AREAI Elself (S= 65)
Then DA = (71.5/ 100)
AREAF = DA*
AREAI Elself (S= 66)
Then DA = (72.5/ 100)
AREAF = DA*
AREAI Elself (S= 67)
Then DA = (73.5 /
100) AREAF = DA *
AREAI Elself (S= 68)
Then DA = (74.5 /
100) AREAF = DA *
AREAI Elself (S= 69)
Then DA = (75.5 /
100) AREAF= DA *
AREAI Elself (S = 70)
Then DA = (76.5 /
100) AREAF = DA *
AREAI Elself (S= 71)
Then DA = (77 / 100)
AREAF = DA *
AREAI Elself (S= 72)
Then DA = (78 / 100)
AREAF = DA*
AREAI Elself (S= 73)
Then
```

```
Elself (S= 96) Then
DA = (96.5 / 100)
AREAF = DA *
AREAI Elself (S= 97)
Then DA = (97.5 /
100) AREAF = DA *
AREAI Elself (S= 98)
Then DA = (98 / 100)
AREAF = DA *
AREAI Elself (S= 99)
Then DA = (99/ 100)
AREAF = DA *
AREAI Elself (S=
100) Then DA = (100/
100) AREAF= DA *
AREAI End If
End Function
```

```
Elself (S= 79) Then DA
= (84/ 100)
AREAF = DA *AREA1
Elself (S= 80) Then DA
= (84.5 / 100) AREAF =
DA * AREA I Elself (S=
81) Then DA = (85 /
100)
AREAF = DA * AREA1
Elself (S= 82) Then DA
= (86/ 100)
AREAF = DA * AREA1
Elself (S= 83) Then DA
= (86.5 / 100) AREAF =
DA * AREA 1 Elself (S=
84) Then DA = (87 /
100)
AREAF = DA * AREA1
Elself (S= 85) Then DA
= (87/ 100)
AREAF = DA * AREA!
Elself (S= 86) Then DA
= (88 / 100)
AREAF = DA *AREA1
Elself (S= 87) Then DA
= (89.5 / 100) AREAF =
DA * AREA1 Elself (S=
88) Then DA = (90/ 100)
AREAF = DA * AREA I
Elself (S= 89) Then DA
= (91 / 100)
AREAF= DA * AREA1
Elself (S= 90) Then DA
= (92/ 100)
AREAF = DA * AREA1
Elself (S= 91) Then DA
= (93 / 100)
AREAF = DA * AREA1
Elself (S= 92) Then DA
= (94/ 100)
AREAF = DA * AREA!
Elself (S= 93) Then DA
= (94.5 / 100) AREAF=
DA * AREA1 Elself (S=
94) Then DA = (95/100)
AREAF = DA * AREA1
Elself (S= 95) Then DA
= (96/ 100)
AREAF = DA * AREA1
Public Function
confirm(a) If (a < 0.21)
Then confirm = 0.21
Else
confirm = a End If
End Function
Public Function
chabnge()
FRMFLOOR.Text5 = 0
FRMFLOOR.Text6 = 0
FRMFLOOR.Text7 = 0
FRM FLOOR.Text8 = 0
FRMFLOOR.Text9 = 0
FRMFLOOR.TextIO = 0
FRMFLOOR.Textl I =0
FRMFLOOR.Textl2 = 0
FRMFLOOR.Textl 3 = 0
FRMFLOOR.Textl4 = 0
FRMFLOOR.Textl 6 = 0
FRMFLOOR.Textl 7 = 0
FRMFLOOR.Textl 8 = 0
FRMFLOOR.Textl9 = 0
FRMFLOOR.Text20 = 0
FRMFLOOR.Text21 =0
FRMFLOOR.Text22 = 0
FRMFLOOR.Text23 = 0
FRMFLOOR.Text24 = 0
FRMFLOOR.Text25 =0
FRMFLOOR.Text26 = 0
FRMFLOOR.Text27 = 0
FRMFLOOR.Text28 = 0
FRMFLOOR.Text29 = 0
FRMFLOOR.Text30 = 0
FRMFLOOR.Text31 =0
FRMFLOOR.Text32 = 0
FRMFLOOR.Text33 =0
FRMFLOOR.Text34 = 0
FRMFLOOR.Text35 = 0
FRMFLOOR.Text36 = 0
FRMFLOOR.Text37 = 0
FRMFLOOR.Text38 =0
FRMFLOOR.Text39 = 0
FRMFLOOR.Text40 = 0
FRMFLOOR.Text41 =0
FRMFLOOR.Text42 = 0
FRMFLOOR.Text43 = 0
FRMFLOOR.Text44 = 0
FRMFLOOR.Text45 = 0
End Function
```

References

Merrit (1973), Building Engineering and System Design, second edition.

Me Graw-Hill (1960), Encyclopedia of Science and Technology, seventh edition.

Norman, C. H. (1987), Modern Air-conditioning Practice, McGraw-Hill Inc.

Ashrae (1985), Fundamentals, American Society of Heating, Refrigeration and Air-conditioning Engineers, Inc. Atlanta.

Carrier (1985), Handbook of Air-conditioning System Design.

Ashrae (1984), Fundamentals, American Society of Heating, Refrigeration and Air-conditioning Engineers, Inc. Atlanta.