Abstract

The theory underlying this project is that it is possible to find pertinent weather data for cities across the Untied States that will allow for the implementation of a program to calculate weather delays on a construction project, specifically, a seven-story fifty-five unit condo mid-rise. After receiving user input for construction times and costs, the program, based on the weather data it has already stored, will compute the number of days the project will be delayed for weather and the total cost that those delays will incur. These are outputted at the end along with the revised finish date that takes into account the time lost because of weather. The user is allowed to see the varying degrees to which weather plays a role in such a construction process as it is attempted in every region of the United States.

Table of contents

Introduction

Project management is a difficult study because it requires a great reliance on estimations and human judgment, rather than concrete information and scientific laws, in order to provide the framework in which a construction project can be completed. Also, when speaking of building construction, project management also necessitates a careful study of weather patterns and probabilities to be able to accurately estimate the time that will be lost due to adverse weather conditions such as heavy rain, snow, or ice, or extremely high or low temperatures. It is essential, therefore, to have a method by which to standardize these calculations, and it is also important to be able to use a computer to solve them, in order to preventing errors in calculation. In short, the problem is how will weather conditions across the United States affect the construction delay time and costs of a mid-rise? To this end, a program has been created to answer just that question. The research process, creation and operation of the program, and conclusions and possible improvements will all hereafter be described.

Methods

Obviously, there could be no hope of creating a functional program without doing significant research to find specific guidelines for developing weather-analysis equations, not to mention the vast amounts of weather data that would have to be stored for input into the program. So, once the topic was decided upon, the research was broken up into two parts. Three teammates began interviewing managers on construction sites and other experts in project management engineering while the fourth collected weather data for the twenty-six cities across the United States. One of the interviewees was the superintendent who actually coordinated the mid-rise construction north of campus that was used as the basis for the program. The relevant weather information was found in internet archives.

Information in hand, the next step was to make it usable within the narrow confines of computer understanding. So, all of the weather data had to be converted into comma-separated data files for the program to call for input and store in matrices. The program also requires user input for the starting month of the project, the amount of time expected to be spent on each phase—groundwork and foundation, structure, exterior, and interior—and the expected cost per day during each of these phases, in thousands of dollars. It is expected that the user will be familiar with such estimates, but for our example we obtained these mainly through conversations had with experts.

The last step is output and analysis. The computer runs through the various algorithms and outputs a list of new data describing the situation. These include the number of rain, snow, and ice delays that can be expected, as well as the total

delays and the corresponding extra cost incurred. In order to solve our problem, we compare the results for different cities and see how climate affects the construction process.

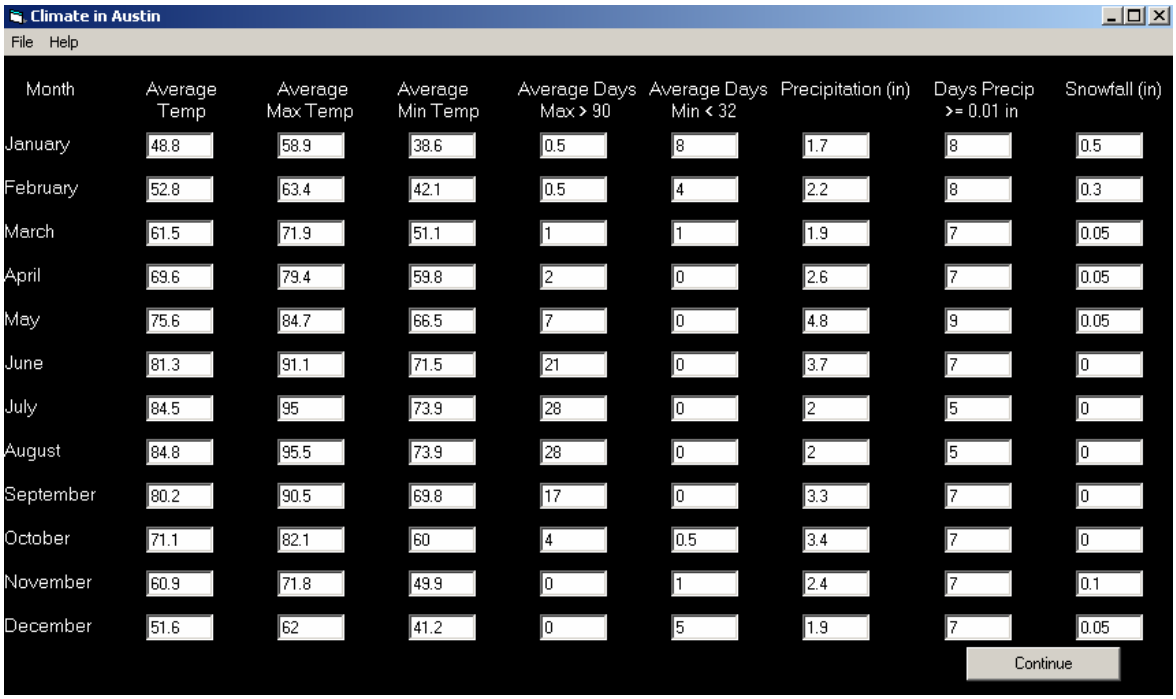
Results

In order to see just how the program is executed, we will run through its stages with hypothetical data.

The first page is a user-interface constructed with a map of the U.S. that allows the user to select a city by clicking on it. And, if the city the user wants is not listed, it is easy to see where the closest one lies that is likely to have similar weather patterns. As one can see, the cities represent all of the major climate regions of the U.S., including Alaska and Hawaii.

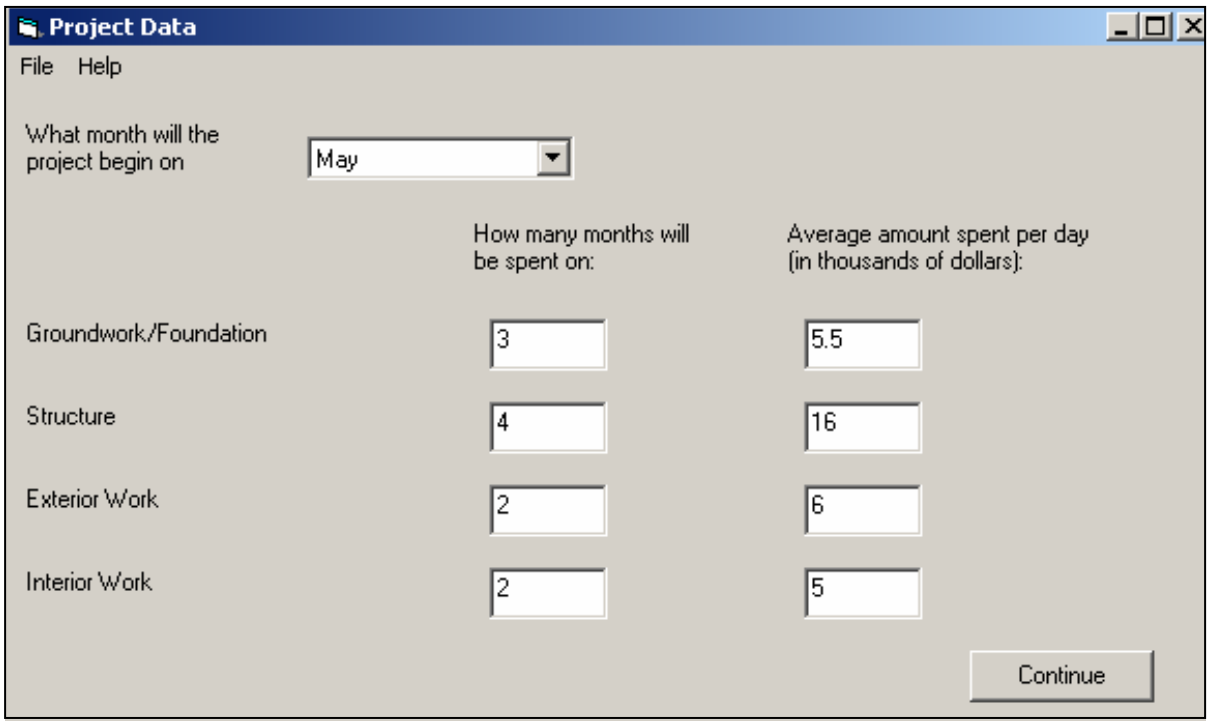
Once the user selects a city (Austin, in this example), a page is displayed with all of that city's weather data in textboxes whose contents can be modified if they are different from data that the user might already have. This allow for varying levels of user input dependent on his expertise and desire for accuracy. The data include all of the following averages for each month of the year: overall temperature, high temperature, low

temperature, days with a high of 90°F or higher, days with a low of 32°F or lower, precipitation, days with precipitation of at least 0.01 inches, and snowfall.



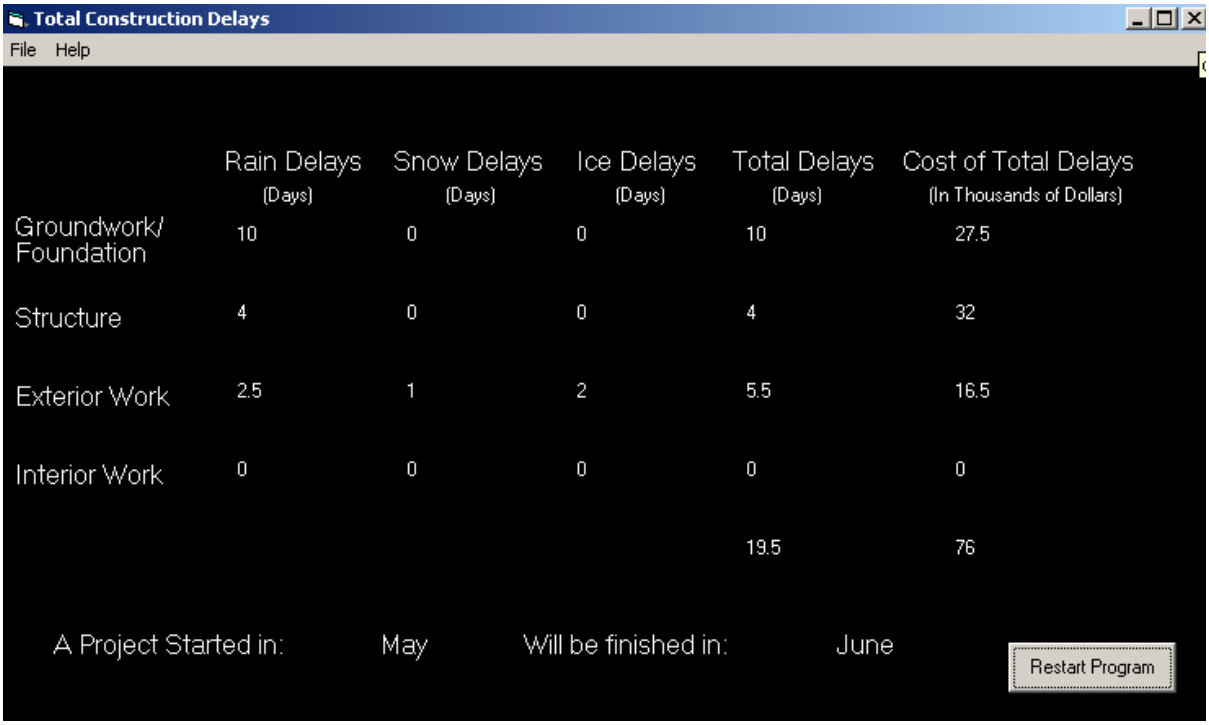
Once the user confirms these numbers by pressing "Continue," the next page is displayed which asks for the information about starting month of the project, time spent in each phase, and cost per day. This is information that the contractor would have, and again, they are critically important to the accuracy of the results. For

instance, a project in Maine that starts in November will have much greater delays than the same project started in April.



For this example, we examine the result of starting in May with the eight values for time and cost that are shown in the figure above. These are the values obtained from conversations and with builders and records of past projects, including the one upon which this program is based. Small changes can produce drastically different results since an extra month spent on the foundation could mean working during a rainy season or during the winter, either of which is likely to cause increased problems and delays.

Again, pressing continue verifies the input and takes the user to the next page.



This last page shows the results of program execution, giving the number of days lost because of each kind of precipitation, and it subdivides these delays into the four basic phases of construction. Adding them to the user-inputted estimates of construction time, it gives the revised finish date for the project. And, in the far right column, the total cost of delay is displayed, which is probably the most important consideration to someone planning a project since resources are almost always in limited supply.

One test for accuracy is to simply compare results to common sense. And, as expected, we see greater snow and ice delays in Alaska than in Florida, and we see greater delays during the foundation process that during interior work.

Conclusions

More than anything, this project was a learning experience. For one, going into the project our team had no idea of the vastness of engineering literature and resources, not did we account for the complexity that quickly presents itself. What we thought was a simple idea soon escalated to a theory worthy of a senior thesis. After just a few hours research, we saw that there is a library of information on the minutest of details of construction and very few simple generalizations about it. The difficulty we then faced was narrowing the topic enough so that we only needed detailed information for one subject. So, we picked weather delays. During the process we learned not only about the effects of weather, but also much about the dynamics of team interaction and contribution, ways to go about seeking professional knowledge, and the convenience or sometimes inconvenience of using Visual Basic to write code.

All in all, our project was fairly successful, given the amount of detail that we were able to include and the algorithms we could find for estimating weather activity. One problem that any such program will face is the unpredictability of weather. What happens one year is certainly not guaranteed to happen the next, and average weather conditions by definition do not allow for the possibility of extremes. These circumstances, of course, limit the accuracy of our project, but we have to base our judgment on a reasonable idea of what is possible. Meteorologists are not deemed ignorant or useless because of an incorrect forecast; likewise this program must be viewed in the same light—sometimes it will be accurate and sometimes it will not.

That being said, there are obvious improvements and extensions that could be made. The cost per day could be broken down into materials and labor and rental costs. From this breakdown, the delay cost per day could be made more accurate by including only labor and rental costs, since the materials will not necessarily go to waste by having to sit an extra day. Also, the program could be extended to allow for a modifiable project where the user could select from a variety of features, from the number of stories and units to the inclusion or exclusion of a parking garage to the materials used for framework. Then, instead of having the user input the average cost per day, the computer would calculate it automatically. These improvements, however, could only be made at a great expense of time and energy in research and a much more complex program code. Such a program has almost certainly already been created and marketed, but ours is still a reasonably effective simplification of such.

References

http://www.climate-zone.com/climate/united-states

Developer: Patty Sanguily with Navona Partners, Austin.

Civil Engineer: Frank Alattar. Houston, Texas.

Architect/Project Manager: Walter Myrick. Houston, Texas.

Jackson, barbara. Construction Management Jumpstart. New York: Wiley, John & Sons, Inc., 2004.

Woolsen, Dodge. Construction Jobsite Management. Chicago: McGraw-Hill Companies, 2004.

Appendix: Visual Basic Source Code

```
Private Sub Command1_Click()
   CityChoice.Show
   Intro.Hide
End Sub

Private Sub mnuExit_Click()
   End
End Sub

Public City As String

Private Sub cmdAlbuquerque_Click()
   City = "Albuquerque"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdAnchorage_Click()
   City = "Anchorage"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdAustin_Click()
   City = "Austin"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdBatonRouge_Click()
   City = "Baton Rouge"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdCharleston_Click()
   City = "Charleston"
   CityClimate.Show
   CityChoice.Hide
```

```
End Sub

Private Sub cmdChicago_Click()
   City = "Chicago"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdColoradoSprings_Click()
   City = "Colorado Springs"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdDallas_Click()
   City = "Dallas"
   CityClimate.Show
   CityChoice.Hide
End Sub


Private Sub cmdDesMoines_Click()
   City = "Des Moines"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdGreatFalls_Click()
   City = "Great Falls"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdHonolulu_Click()
   City = "Honolulu"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdHuntsville_Click()
   City = "Huntsville"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdLasVegas_Click()
```

```
    City = "Las Vegas"
    CityClimate.Show
    CityChoice.Hide
End Sub

Private Sub cmdLosAngeles_Click()
    City = "Los Angeles"
    CityClimate.Show
    CityChoice.Hide
End Sub

Private Sub cmdLouisville_Click()
    City = "Louisville"
    CityClimate.Show
    CityChoice.Hide
End Sub

Private Sub cmdMemphis_Click()
    City = "Memphis"
    CityClimate.Show
    CityChoice.Hide
End Sub

Private Sub cmdMinneapolis_Click()
    City = "Minneapolis"
    CityClimate.Show
    CityChoice.Hide
End Sub

Private Sub cmdNewYork_Click()
    City = "New York City"
    CityClimate.Show
    CityChoice.Hide
End Sub

Private Sub cmdOklahomaCity_Click()
    City = "Oklahoma City"
    CityClimate.Show
    CityChoice.Hide
End Sub

Private Sub cmdOrlando_Click()
    City = "Orlando"
    CityClimate.Show
    CityChoice.Hide
End Sub
```

```
Private Sub cmdPortland_Click()
   City = "Portland"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdSacramento_Click()
   City = "Sacramento"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdSalem_Click()
   City = "Salem"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdSeattle_Click()
   City = "Seattle"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub cmdWashington_Click()
   City = "Washington D.C."
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub mnuExit_Click()
   End
End Sub

Private Sub cmdSaltLakeCity_Click()
   City = "Salt Lake City"
   CityClimate.Show
   CityChoice.Hide
End Sub

Private Sub mnuHelp_Click()
   MsgBox ("Choose a city from those listed on the map." & Chr(13) & Chr(10) & "A
climate profile will be returned for that city.")
End Sub
```

```
Private Sub mnuRestart_Click()
   CityChoice.Hide
   Intro.Show
End Sub

Public City As String
Private Function RainDelay(precip, hiPrecip) As Integer
   If precip / hiPrecip >= 1 Then
      RainDelay = hiPrecip
   ElseIf precip / hiPrecip > 0.5 Then
      RainDelay = hiPrecip / 2
   Else
      RainDelay = hiPrecip / 3
   End If
End Function
Private Function SnowDelay(snow) As Integer
   If snow >= 5 Then
      SnowDelay = 20
   ElseIf snow >= 2 Then
      SnowDelay = 10
   ElseIf snow >= 0.5 Then
      SnowDelay = 2
   Else
      SnowDelay = 0
   End If
End Function
Private Function IceDelay(tempLess, hiPrecip) As Integer
   If tempLess / hiPrecip >= 2 Then
      If tempLess > 15 Then
         IceDelay = hiPrecip
      Else
         IceDelay = hiPrecip / 2
      End If
   ElseIf tempLess / hiPrecip >= 1 Then
      If tempLess > 15 Then
         IceDelay = hiPrecip / 2
      Else
         IceDelay = hiPrecip / 4
      End If
   Else
      IceDelay = 0
   End If
End Function
Private Sub Command1_Click()
   Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Delay.txt" For
Output As #2
```

```
    For k = 0 To 11
        tempLess = txtLess(k).Text
        precip = txtPrecip(k).Text
        hiPrecip = txtHiPrecip(k).Text
        snow = txtSnow(k).Text
        rDelay = RainDelay(precip, hiPrecip)
        sDelay = SnowDelay(snow)
        iDelay = IceDelay(tempLess, hiPrecip)
        Write #2, rDelay, sDelay, iDelay
    Next k
    Close #2
    CityClimate.Hide
    ProjectInput.Show
End Sub

Private Sub Form_Load()
    City = CityChoice.City
    CityClimate.Caption = "Climate in " & City
    Select Case City
        Case Is = "Albuquerque"
        Open "C:\Documents and Settings\csp253\My
Documents\Labs\Project\Albuquerque.txt" For Input As #1
        Case Is = "Anchorage"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Anchorage.txt"
For Input As #1
        Case Is = "Austin"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Austin.txt" For
Input As #1
        Case Is = "Baton Rouge"
        Open "C:\Documents and Settings\csp253\My
Documents\Labs\Project\BatonRouge.txt" For Input As #1
        Case Is = "Charleston"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Charleston.txt"
For Input As #1
        Case Is = "Chicago"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Chicago.txt"
For Input As #1
        Case Is = "Colorado Springs"
        Open "C:\Documents and Settings\csp253\My
Documents\Labs\Project\ColoradoSprings.txt" For Input As #1
        Case Is = "Dallas"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Dallas.txt" For
Input As #1
        Case Is = "Des Moines"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\DesMoines.txt"
For Input As #1
```

```
        Case Is = "Great Falls"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\GreatFalls.txt"
For Input As #1
        Case Is = "Honolulu"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Honolulu.txt"
For Input As #1
        Case Is = "Huntsville"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Huntsville.txt"
For Input As #1
        Case Is = "Las Vegas"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\LasVegas.txt"
For Input As #1
        Case Is = "Los Angeles"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\LosAngeles.txt"
For Input As #1
        Case Is = "Louisville"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Louisville.txt"
For Input As #1
        Case Is = "Memphis"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Memphis.txt"
For Input As #1
        Case Is = "Mineapolis"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Mineapolis.txt"
For Input As #1
        Case Is = "New York City"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\NewYork.txt"
For Input As #1
        Case Is = "Oklahoma City"
        Open "C:\Documents and Settings\csp253\My
Documents\Labs\Project\OklahomaCity.txt" For Input As #1
        Case Is = "Orlando"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Orlando.txt"
For Input As #1
        Case Is = "Portland"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Portland.txt"
For Input As #1
        Case Is = "Sacramento"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Sacramento.txt"
For Input As #1
        Case Is = "Salem"
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Salem.txt" For
Input As #1
        Case Is = "Salt Lake City"
        Open "C:\Documents and Settings\csp253\My
Documents\Labs\Project\SaltLakeCity.txt" For Input As #1
        Case Is = "Seattle"
```

```
        Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Seattle.txt" For
Input As #1
        Case Is = "Washington D.C."
        Open "C:\Documents and Settings\csp253\My
Documents\Labs\Project\Washington.txt" For Input As #1
    End Select

    For k = 0 To 11
        Input #1, avgTemp, avgMax, avgMin, tempGreater, tempLess, precip, hiPrecip, snow
        txtAvgTemp(k).Text = avgTemp
        txtMaxTemp(k).Text = avgMax
        txtMinTemp(k).Text = avgMin
        txtGreater(k).Text = tempGreater
        txtLess(k).Text = tempLess
        txtPrecip(k).Text = precip
        txtHiPrecip(k).Text = hiPrecip
        txtSnow(k).Text = snow
    Next k
    Close #1
End Sub

Private Sub mnuBack_Click()
    CityChoice.Show
    CityClimate.Hide
End Sub

Private Sub mnuExit_Click()
    End
End Sub

Private Sub mnuHelp_Click()
    MsgBox ("This is a climate profile for the selected city." & Chr(13) & Chr(10) &
"Temperatures are in degrees Fahrenheit and precipitation is measured in inches." & Chr(13)
& Chr(10) & "You may edit the values for a more specific climate pattern.")
End Sub

Private Sub mnuRestart_Click()
    CityClimate.Cls
    CityClimate.Hide
    Intro.Show
End Sub

Public startMonth As String
Public monthGWF As Integer, monthS As Integer, monthEW As Integer, monthIW As
Integer
Public costGWF As Single, costS As Single, costEW As Single, costIW As Single
```

```
Private Sub cmdContinue_Click()
   startMonth = cboStartMonth.Text
   monthGWF = txtMonthsGWF.Text
   monthS = txtMonthsS.Text
   monthEW = txtMonthsEW.Text
   monthIW = txtMonthsIW.Text
   costGWF = txtCostGWF.Text
   costS = txtCostS.Text
   costEW = txtCostEW.Text
   costIW = txtCostIW.Text
   ProjectInput.Hide
   DelayTotals.Show
End Sub

Private Sub mnuBack_Click()
   ProjectInput.Hide
   CityClimate.Show
End Sub

Private Sub mnuExit_Click()
   End
End Sub

Private Sub mnuHelp_Click()
   MsgBox ("Enter predicted construction information.")
End Sub

Private Sub mnuRestart_Click()
   ProjectInput.Cls
   ProjectInput.Hide
   CityClimate.Cls
   Intro.Show
End Sub

Private Sub cmdRestart_Click()
   DelayTotals.Cls
   DelayTotals.Hide
   ProjectInput.Cls
   CityClimate.Cls
   CityChoice.Show
End Sub

Private Sub Form_Load()
   Open "C:\Documents and Settings\csp253\My Documents\Labs\Project\Delay.txt" For
Input As #2
```

```
    Dim Delays(11, 2) As Integer
    Dim startMonth As String
    startMonth = ProjectInput.startMonth
    lblStartMonth.Caption = startMonth
    Dim monthGWF As Integer, monthS As Integer, monthEW As Integer, monthIW As
Integer
    monthGWF = ProjectInput.monthGWF
    monthS = ProjectInput.monthS
    monthEW = ProjectInput.monthEW
    monthIW = ProjectInput.monthIW
    Dim costGWF As Single, costS As Single, costEW As Single, costIW As Single
    costGWF = ProjectInput.costGWF
    costS = ProjectInput.costS
    costEW = ProjectInput.costEW
    costIW = ProjectInput.costIW

    For k = 0 To 11
       Input #2, rain, snow, ice
       Delays(k, 0) = rain
       Delays(k, 1) = snow
       Delays(k, 2) = ice
     Next k

     Select Case startMonth
       Case Is = "January"
          n = 0
       Case Is = "February"
          n = 1
       Case Is = "March"
          n = 2
       Case Is = "April"
          n = 3
       Case Is = "May"
          n = 4
       Case Is = "June"
          n = 5
       Case Is = "July"
          n = 6
       Case Is = "August"
          n = 7
       Case Is = "September"
          n = 8
       Case Is = "October"
          n = 9
       Case Is = "November"
          n = 10
```

```
Case Is = "December"
   n = 11
End Select

Dim rainD As Integer, snowD As Integer, iceD As Integer
rainD = 0
snowD = 0
iceD = 0

i = n + monthGWF
If i > 11 Then
   i = i - 12
End If
k = n
Do
   rainD = rainD + Delays(k, 0)
   snowD = snowD + Delays(k, 1)
   iceD = iceD + Delays(k, 2)
   k = k + 1
   If k = 12 Then
      k = 0
   End If
 Loop Until k = i
 lblRain(0).Caption = rainD
 lblSnow(0).Caption = snowD
 lblIce(0).Caption = iceD
 totalGWF = rainD + snowD + iceD
 lblTot(0).Caption = totalGWF
 lblCost(0) = totalGWF * costGWF / 2
 n = k + CInt(totalGWF / 30)
 If n > 11 Then
   n = n - 12
 End If

  rainD = 0
  snowD = 0
  iceD = 0

  i = n + monthS
  If i > 11 Then
    i = i - 12
  End If
  k = n
Do
   rainD = rainD + Delays(k, 0)
   snowD = snowD + Delays(k, 1)
```

```
    iceD = iceD + Delays(k, 2)
    k = k + 1
    If k = 12 Then
       k = 0
    End If
  Loop Until k = i
   lblRain(1).Caption = rainD / 2
   lblSnow(1).Caption = snowD
   lblIce(1).Caption = iceD
   totalS = rainD / 2 + snowD + iceD
   lblTot(1).Caption = totalS
   lblCost(1) = totalS * costS / 2
   n = k + CInt((totalGWF + totalS) / 30)
   If n > 11 Then
     n = n - 12
   End If

  rainD = 0
  snowD = 0
  iceD = 0

  i = n + monthEW
  If i > 11 Then
     i = i - 12
  End If
 k = n
Do
    rainD = rainD + Delays(k, 0)
    snowD = snowD + Delays(k, 1)
    iceD = iceD + Delays(k, 2)
    k = k + 1
    If k = 12 Then
       k = 0
    End If
  Loop Until k = i
   lblRain(2).Caption = rainD / 2
   lblSnow(2).Caption = snowD / 2
   lblIce(2).Caption = iceD
   totalEW = rainD / 2 + snowD / 2 + iceD
   lblTot(2).Caption = totalEW
   lblCost(2) = totalEW * costEW / 2
   n = k + CInt((totalGWF + totalS + totalEW) / 30)
   If n > 11 Then
     n = n - 12
   End If
```

```
        rainD = 0
        snowD = 0
        iceD = 0

        i = n + monthIW
        If i > 11 Then
           i = i - 12
        End If
        k = n
       Do
          rainD = rainD + Delays(k, 0)
          snowD = snowD + Delays(k, 1)
          iceD = iceD + Delays(k, 2)
          k = k + 1
          If k = 12 Then
             k = 0
          End If
       Loop Until k = i
        lblRain(3).Caption = 0
        lblSnow(3).Caption = snowD
        lblIce(3).Caption = iceD
        totalIW = snowD / 2 + iceD
        lblTot(3).Caption = totalIW
        lblCost(3) = totalIW * costIW / 2
        n = k + CInt((totaIW + totalGWF + totalS + totalEW) / 30)
        If n > 11 Then
          n = n - 12
        End If

        lblTot(4).Caption = totalIW + totalEW + totalS + totalGWF
        lblCost(4) = (totalIW * costIW + totalEW * costEW + totalS * costS + totalGWF *
costGWF) / 2

       Select Case n
       Case Is = 0
          finishMonth = "January"
       Case Is = 1
          finishMonth = "February"
       Case Is = 2
          finishMonth = "March"
       Case Is = 3
          finishMonth = "April"
       Case Is = 4
          finishMonth = "May"
       Case Is = 5
          finishMonth = "June"
```

```
        Case Is = 6
           finishMonth = "July"
        Case Is = 7
           finishMonth = "August"
        Case Is = 8
           finishMonth = "September"
        Case Is = 9
           finishMonth = "October"
        Case Is = 10
           finishMonth = "November"
        Case Is = 11
           finishMonth = "December"
        End Select

        lblFinishMonth.Caption = finishMonth
        Close #2
End Sub

Private Sub mnuBack_Click()
   DelayTotals.Cls
   DelayTotals.Hide
   ProjectInput.Show
End Sub

Private Sub mnuExit_Click()
   End
End Sub

Private Sub mnuHelp_Click()
   MsgBox ("Total delays given in days and delay costs given in thousands of dollars")
End Sub

Private Sub mnuRestart_Click()
   DelayTotals.Cls
   DelayTotals.Hide
   ProjectInput.Cls
   CityClimate.Cls
   Intro.Show
End Sub
```