```
' Kevin Fox
' Programming User Interfaces
' Assignment 2
' February 4, 2003


Option Explicit

' Had a little trouble with constants and declaring globals, even
' after the TA emails, so I went with regular variables, initialized
' in Form_Load()

Dim Floors As Integer
Dim Elevators As Integer
Dim ElevatorFloor(3) As Integer
Dim FloorHeight As Integer


' Is the passed floor waiting for an elevator?
Private Function IsActive(Index As Integer)
  'Active state is kept as a boolean in the floor's tag.
  IsActive = FloorCall.Item(Index).Tag
End Function


' Are any floors above the passed floor waiting for an elevator?
Private Function CheckUp(Index As Integer)
  Dim i As Integer
  Dim active As Boolean

  CheckUp = False
  If Index < Floors - 1 Then
    For i = Index + 1 To Floors - 1
      active = IsActive(i)
      If active Then
        CheckUp = True
        Exit Function
      End If
    Next
  End If
End Function


' Are any floors below the passed floor waiting for an elevator?
Private Function CheckDown(Index As Integer)
  Dim i As Integer
  Dim active As Boolean

  CheckDown = False
  If Index > 0 Then
    For i = 0 To Index - 1
      active = IsActive(i)
      If active Then
        CheckDown = True
        Exit Function
      End If
    Next
  End If
```

```
End Function


' Set a floor to inactive, whether it was active or inactive before.
Private Sub Extinguish(Index As Integer)
  FloorCall.Item(Index).Tag = False
  FloorCall.Item(Index).BackColor = RGB(10, 10, 10)
End Sub


' Move the passed elevator up one floor
Private Sub ElevatorUp(ElevIndex As Integer)
  Elevator.Item(ElevIndex).Top = Elevator.Item(ElevIndex).Top -
FloorHeight
  ElevatorFloor(ElevIndex) = ElevatorFloor(ElevIndex) + 1
End Sub


' Move the passed elevator down one floor
Private Sub ElevatorDown(ElevIndex As Integer)
  Elevator.Item(ElevIndex).Top = Elevator.Item(ElevIndex).Top +
FloorHeight
  ElevatorFloor(ElevIndex) = ElevatorFloor(ElevIndex) - 1
End Sub


' Change the elevator's direction (stored in the tag boolean)
' Also change elevator icon to match.
Private Sub ChangeDirection(Index As Integer)
  If Elevator.Item(Index).Tag Then
    Elevator.Item(Index).Picture = DownPic.Picture
  Else
    Elevator.Item(Index).Picture = UpPic.Picture
  End If
  If Elevator.Item(Index).Tag Then
    Elevator.Item(Index).Tag = False
  Else
    Elevator.Item(Index).Tag = True
  End If
End Sub


' Using current elevator state (of the elevator index passed to
' the function) and the floor states, Elevator_Brain contains the
' move logic, and executes the move and floor resetting.
Private Sub Elevator_Brain(Index As Integer)
  Dim goingUp As Boolean

  goingUp = Elevator.Item(Index).Tag
  If goingUp And CheckUp(ElevatorFloor(Index)) Then
    ElevatorUp (Index)
  ElseIf Not goingUp And CheckDown(ElevatorFloor(Index)) Then
    ElevatorDown (Index)
  Else
    ChangeDirection (Index)
  End If

  ' Wherever the elevator ends up, that floor gets deactivated.
```

```
  Extinguish (ElevatorFloor(Index))
End Sub

' Mark a floor as active, both in its tag boolean and visually
Private Sub CallButton_Click(Index As Integer)
  FloorCall.Item(Index).Tag = True
  FloorCall.Item(Index).BackColor = RGB(255, 245, 136)
End Sub

Private Sub Form_Load()
  Dim i As Integer

  ' Initialize parameters. (Ideally I'd use constants here)
  Floors = 10
  Elevators = 3
  FloorHeight = 615
  Timer1.Tag = 0

  For i = 0 To Elevators - 1
    ElevatorFloor(i) = 4
    Elevator.Item(i).Tag = False
  Next

  For i = 0 To Floors - 1
    FloorCall.Item(i).Tag = False
    FloorCall.Item(i).BackColor = RGB(10, 10, 10)
  Next

End Sub

' Every second, rotate the active elevator and call its brain.
Private Sub Timer1_Timer()
  If Timer1.Tag = Elevators Then Timer1.Tag = 0
  Elevator_Brain (Timer1.Tag)
  Timer1.Tag = Timer1.Tag + 1
End Sub
```