Dim XRes As Integer, YRes As Integer, cState As String
Dim Pic(600, 600, 2) As Integer, PicTemp(600, 600, 2) As Integer
Dim Part1 As Integer, Part2 As Integer

Private Type ColRGB
    Red As Integer
    Green As Integer
    Blue As Integer
End Type

Event Resize(Width As Long, Height As Long)
Event Progress(Prog As Integer, State As String)
Const P1Val = 75
Const P2Val = 100 - P1Val
Const S1 = "Reading File:")
Const S2 = "Analyzing File:")
Const S3 = "Drawing Image:")
Const S4 = "Done"

Private Function LoadFile(Picture As PictureBox)
    On Error Resume Next
    Dim Counter1 As Long, Counter2 As Long, Val As Long
    Timer.Enabled = True
    Part1 = 0
    Part2 = 0
    cState = S1
    Picture.ScaleMode = 3
    XRes = Picture.ScaleWidth
    YRes = Picture.ScaleHeight
    For Counter1 = 1 To YRes
        For Counter2 = 1 To XRes
            Val = Picture.Point(Counter2, Counter1)
            Pic(Counter2, Counter1, 0) = GetRGB(Val).Red
            Pic(Counter2, Counter1, 1) = GetRGB(Val).Green
            Pic(Counter2, Counter1, 2) = GetRGB(Val).Blue
        Next Counter2
        DoEvents
    Next Counter1
    cState = S2
End Function

Function OpenPicture(Filename As String)
    On Error Resume Next
    Target.Picture = LoadPicture(Filename)
End Function

Private Function GetRGB(ByVal Color As Long) As ColRGB
    On Error Resume Next
    GetRGB.Red = Color Mod &H100
    GetRGB.Green = (Color \
        &H100) Mod &H100
    GetRGB.Blue = (Color \ 
        &H10000) Mod &H100
End Function

Function invert()
    On Error Resume Next
    Dim VPic(2) As Integer, i As Integer, j As Integer, k As Integer
    LoadFile Target
    For i = 0 To YRes
        For j = 0 To XRes
            For k = 0 To 2
                VPic(2 - k) = 255 - Pic(j, i, k)
            Next k
        Next j
    Next i
End Function
PicTemp(j, i, k) = 255 - Pic(j, i, k)
Next
Next
Part1 = Int((i / YRes) * P1Val)
DoEvents
Next
Draw Target
End Function

Function diffuse()
On Error Resume Next
Dim RndX As Integer, RndY As Integer, i As Integer, j As Integer,
DiffuseVal As Integer
DiffuseVal = GetVal(1, 25, 1, "Diffuse")
If DiffuseVal = 1000 Then Exit Function
LoadFile Target
For i = DiffuseVal To YRes - DiffuseVal
    For j = DiffuseVal To XRes - DiffuseVal
        RndX = Rnd * DiffuseVal
        RndY = Rnd * DiffuseVal
        PicTemp(j, i, 0) = Pic(j + RndX, i + RndY, 0)
        PicTemp(j, i, 1) = Pic(j + RndX, i + RndY, 1)
        PicTemp(j, i, 2) = Pic(j + RndX, i + RndY, 2)
    Next
    Part1 = Int((i / YRes) * P1Val)
    DoEvents
Next
Draw Target
End Function

Function emboss()
On Error Resume Next
Dim VPic As Integer, EmbossVal As Integer
EmbossVal = GetVal(1, 10, 1, "Emboss")
If EmbossVal = 1000 Then Exit Function
LoadFile Target
For i = EmbossVal To YRes - EmbossVal
    For j = EmbossVal To XRes - EmbossVal
        For k = 0 To 2
            VPic = Pic(j, i, k) - Pic(j - EmbossVal, i - EmbossVal, k) + 128
            If VPic < 0 Then VPic = 0
            If VPic > 255 Then VPic = 255
            PicTemp(j, i, k) = VPic
        Next
    Next
    Part1 = Int((i / YRes) * P1Val)
    DoEvents
Next
Draw Target
End Function

Function smooth()
On Error Resume Next
Dim i As Integer, j As Integer, VPic(2) As Integer
Dim k As Integer, l As Integer, m As Integer, SmoothVal As Integer
SmoothVal = GetVal(1, 10, 1, "Smooth")
If SmoothVal = 1000 Then Exit Function
SmoothVal = SmoothVal + 2
LoadFile Target
For i = Int(SmoothVal / 2) To YRes - Int(SmoothVal / 2)
    For j = Int(SmoothVal / 2) To XRes - Int(SmoothVal / 2)
        For k = -Int(SmoothVal / 2) To Int(SmoothVal / 2)
            For l = -Int(SmoothVal / 2) To Int(SmoothVal / 2)
                For m = 0 To 2
                    VPic(m) = VPic(m) + Pic(j + k, i + l, m)
                Next
            Next
        Next
    Next
Next
For m = 0 To 2
    VPic(m) = VPic(m) / SmoothVal ^ 2
Next
For k = 0 To 2
    PicTemp(j, i, k) = VPic(k)
Next
Part1 = Int((i / YRes) * P1Val)
DoEvents
Next
Draw Target
End Function

Function sharpen()
On Error Resume Next
Dim SharpenPercent As Double, VPic(2) As Integer, SharpenVal As Integer
SharpenVal = GetVal(1, 500, 1, "Sharpen")
If SharpenVal = 1000 Then Exit Function
SharpenVal = SharpenVal + 1
LoadFile Target
SharpenPercent = SharpenVal / 10
For i = 1 To YRes
    For j = 1 To XRes
        For k = 0 To 2
            VPic(k) = Pic(j, i, k) + SharpenPercent * (Pic(j, i, k) - Pic(j - 1, i - 1, k))
            If VPic(k) < 0 Then VPic(k) = 0
            If VPic(k) > 255 Then VPic(k) = 255
            PicTemp(j, i, k) = VPic(k)
        Next
    Next
    Part1 = Int((i / YRes) * P1Val)
    DoEvents
Next
Draw Target
End Function

Function engrave()
On Error Resume Next
Dim VPic(2) As Integer, bRelX As Integer, bRelY As Integer,
EngraveVal As Integer
EngraveVal = GetVal(1, 10, 1, "Engrave")
If EngraveVal = 1000 Then Exit Function
LoadFile Target
For i = EngraveVal To YRes - EngraveVal
    For j = EngraveVal To XRes - EngraveVal
        For k = 0 To 2
            VPic(k) = Pic(j, i, k) - Pic(j + EngraveVal, i + EngraveVal, k) + 128
If VPic(k) < 0 Then VPic(k) = 0
If VPic(k) > 255 Then VPic(k) = 255
PicTemp(j, i, k) = VPic(k)
Next
Next
Part1 = Int((i / YRes) * P1Val)
DoEvents
Next
Next
Draw Target
End Function

Function mosaic()
On Error Resume Next
Dim sMosaic As Integer, VPic As Integer, MosaicVal As Integer
MosaicVal = GetVal(1, 50, 0, "Mosaic")
If MosaicVal = 1000 Then Exit Function
LoadFile Target
MosaicVal = MosaicVal + 2
BlockSize = MosaicVal * MosaicVal
For i = 0 To YRes - MosaicVal Step MosaicVal
   For j = 0 To XRes - MosaicVal Step MosaicVal
      mr = 0: mg = 0: mb = 0
      For k1 = 0 To MosaicVal
         For k2 = 0 To MosaicVal
            mr = mr + Pic(j + k1, i + k2, 0)
            mg = mg + Pic(j + k1, i + k2, 1)
            mb = mb + Pic(j + k1, i + k2, 2)
         Next
      Next
      mr = mr / BlockSize
      mg = mg / BlockSize
      mb = mb / BlockSize
      For k1 = 0 To MosaicVal
         For k2 = 0 To MosaicVal
            PicTemp(j + k2, i + k1, 0) = mr
            PicTemp(j + k2, i + k1, 1) = mg
            PicTemp(j + k2, i + k1, 2) = mb
         Next
      Next
      sMosaic = 0
   Next
   Part1 = Int((i / YRes) * P1Val)
   DoEvents
Next
Next
Draw Target
End Function

Function midday()
On Error Resume Next
Dim sMosaic As Integer, VPic As Integer, MosaicVal As Integer
MosaicVal = GetVal(1, 4, 1, "Midday")
If MosaicVal = 1000 Then Exit Function
LoadFile Target
MosaicVal = 5 - MosaicVal
BlockSize = MosaicVal * MosaicVal
For i = 0 To YRes - MosaicVal
   For j = 0 To XRes - MosaicVal
      mr = 0: mg = 0: mb = 0
      For k1 = 0 To MosaicVal
         For k2 = 0 To MosaicVal
            PicTemp(j + k2, i + k1, 0) = mr
            PicTemp(j + k2, i + k1, 1) = mg
            PicTemp(j + k2, i + k1, 2) = mb
         Next
      Next
      Draw Target
   End Function
mr = mr + Pic(j + k1, i + k2, 0)
mg = mg + Pic(j + k1, i + k2, 1)
mb = mb + Pic(j + k1, i + k2, 2)
Next
mr = mr / BlockSize
mg = mg / BlockSize
mb = mb / BlockSize
For k1 = 0 To MosaicVal
  For k2 = 0 To MosaicVal
    PicTemp(j + k2, i + k1, 0) = mr
    PicTemp(j + k2, i + k1, 1) = mg
    PicTemp(j + k2, i + k1, 2) = mb
  Next
Next
sMosaic = 0
Next
Part1 = Int((i / YRes) * P1Val)
DoEvents
Next
Draw Target
End Function

Function neon()
  On Error Resume Next
  Dim VPic(2) As Integer
  LoadFile Target
  For i = 0 To YRes
    For j = 0 To XRes
      For k = 0 To 2
        g1 = (Pic(j, i, k) - Pic(j + 1, i, k)) ^ 2
        g2 = (Pic(j, i, k) - Pic(j, i + 1, k)) ^ 2
        VPic(k) = 2 * (g1 + g2) ^ 0.5
        If VPic(k) > 255 Then VPic(k) = 255
        PicTemp(j, i, k) = VPic(k)
      Next
    Next
    Part1 = Int((i / YRes) * P1Val)
    DoEvents
  Next
  Draw Target
End Function

Function ColourToMono()
  On Error Resume Next
  Dim GreyVal As Integer
  LoadFile Target
  For i = 0 To YRes
    For j = 0 To XRes
      GreyVal = 0.3 * Pic(j, i, 0) + 0.59 * Pic(j, i, 1) + 0.11 * Pic(j, i, 2)
      For k = 0 To 2
        PicTemp(j, i, k) = GreyVal
      Next
    Next
    Part1 = Int((i / YRes) * P1Val)
    DoEvents
  Next
  Draw Target
End Function
Function brighten()
On Error Resume Next
Dim Val As Integer, R As Integer, G As Integer, b As Integer
Val = GetVal(-255, 255, 0, "Brighten")
If Val = 1000 Then Exit Function
LoadFile Target
For i = 0 To YRes
    For j = 0 To XRes
        R = Pic(j, i, 0) + Val
        If R < 0 Then R = 0
        G = Pic(j, i, 1) + Val
        If G < 0 Then G = 0
        b = Pic(j, i, 2) + Val
        If b < 0 Then b = 0
        PicTemp(j, i, 0) = R
        PicTemp(j, i, 1) = G
        PicTemp(j, i, 2) = b
    Next
    Part1 = Int((i / YRes) * P1Val)
    DoEvents
Next
Draw Target
End Function

Function EditRed()
On Error Resume Next
Dim Val As Integer, R As Integer, G As Integer, b As Integer
Val = GetVal(-255, 255, 0, "Edit Red")
If Val = 1000 Then Exit Function
LoadFile Target
For i = 0 To YRes
    For j = 0 To XRes
        R = Pic(j, i, 0) + Val
        If R < 0 Then R = 0
        G = Pic(j, i, 1)
        b = Pic(j, i, 2)
        PicTemp(j, i, 0) = R
        PicTemp(j, i, 1) = G
        PicTemp(j, i, 2) = b
    Next
    Part1 = Int((i / YRes) * P1Val)
    DoEvents
Next
Draw Target
End Function

Function EditBlue()
On Error Resume Next
Dim Val As Integer, R As Integer, G As Integer, b As Integer
Val = GetVal(-255, 255, 0, "Edit Blue")
If Val = 1000 Then Exit Function
LoadFile Target
For i = 0 To YRes
    For j = 0 To XRes
        R = Pic(j, i, 0)
        G = Pic(j, i, 1)
        b = Pic(j, i, 2) + Val
        If b < 0 Then b = 0
Private Function Draw()
On Error Resume Next
Dim R As Integer, G As Integer, b As Integer
SetUndo
CState = S3
Target.Cls
For i = 0 To YRes
For j = 0 To XRes
R = PicTemp(j, i, 0)
G = PicTemp(j, i, 1)
b = PicTemp(j, i, 2)
Target.PSet (j, i), RGB(R, G, b)
Next
Part2 = Int((i / YRes) * P2Val)
DoEvents
Next
Part1 = P1Val
Part2 = P2Val
CState = S4
RaiseEvent Progress(100, CState)
Timer.Enabled = False
UserControl.Width = UserControl.Width + 1
UserControl.Width = UserControl.Width - 1
End Function

Private Function GetVal(MinVal As Integer, MaxVal As Integer, Val As Integer, Title As String) As Integer
On Error Resume Next
    Val = GetVal(-255, 255, 0, "Edit Green")
If Val = 1000 Then Exit Function
End Function
SelVal.ProgMax = MaxVal
SelVal.ProgMin = MinVal
SelVal.ProgVal = Val
SelVal.EditVal = Title
SelVal.Show 1
Do While SelVal.Visible = True
    DoEvents
    Loop
GetVal = SelVal.ProgVal
DoEvents
End Function

Private Sub Target_Resize()
    On Error Resume Next
    UserControl.Width = Target.Width
    UserControl.Height = Target.Height
    RaiseEvent Resize(Target.Width, Target.Height)
End Sub

Private Function SetUndo()
    On Error Resume Next
    UndoPic.Picture = Target.Image
    DoEvents
End Function

Function undo()
    On Error Resume Next
    Target.Picture = UndoPic.Image
End Function

Private Sub Timer_Timer()
    Dim tVal As Integer
tVal = Part1 + Part2
    RaiseEvent Progress(tVal, cState)
End Sub

Function tonebalance()
    On Error Resume Next
    Dim MidVal As Integer
    LoadFile Target
    For i = 0 To 255
        BalVal(i) = 0
    Next
    For i = 0 To YRes
        For j = 0 To XRes
            MidVal = 0.3 * Pic(j, i, 0) + 0.59 * Pic(j, i, 1) + 0.11 * Pic(j, i, 2)
            BalVal(MidVal) = BalVal(MidVal) + 1
            DoEvents
        Next
    Next
    SetVal
End Function

Function sShift(Min As Integer, Max As Integer, Val As Integer)
    On Error Resume Next
    Dim MidVal As Integer, tVal As Integer
    LoadFile Target
    For i = 0 To YRes
        For j = 0 To XRes
            MidVal = 0.3 * Pic(j, i, 0) + 0.59 * Pic(j, i, 1) + 0.11 * Pic(j, i, 2)
            BalVal(MidVal) = BalVal(MidVal) + 1
            DoEvents
        Next
    Next
    SetVal
End Function
MidVal = 0.3 * Pic(j, i, 0) + 0.59 * Pic(j, i, 1) + 0.11 * Pic(j, i, 2)

If MidVal >= Min And MidVal <= Max Then
    tVal = Pic(j, i, 0) - Val
    If tVal < 0 Then tVal = 0
    PicTemp(j, i, 0) = tVal

    tVal = Pic(j, i, 1) - Val
    If tVal < 0 Then tVal = 0
    PicTemp(j, i, 1) = tVal

    tVal = Pic(j, i, 2) - Val
    If tVal < 0 Then tVal = 0
    PicTemp(j, i, 2) = tVal
Else
    PicTemp(j, i, 0) = Pic(j, i, 0)
    PicTemp(j, i, 1) = Pic(j, i, 1)
    PicTemp(j, i, 2) = Pic(j, i, 2)
End If
Next
Next
Draw
End Function

Function SetVal()
    Graph.Show 1
    Do While Graph.Visible = True
        DoEvents
    Loop
    If Graph.MaxVal = -1 Then Exit Function
    sShift Graph.MinVal, Graph.MaxVal, Graph.Val
End Function

Function ColourBalance()
On Error Resume Next
    LoadFile Target
    For j = 1 To 3
        For i = 0 To 255
            ColVal(j, i) = 0
        Next
    Next

    For i = 0 To YRes
        For j = 0 To XRes
            For k = 1 To 3
                ColVal(k, Pic(j, i, k - 1)) = ColVal(k, Pic(j, i, k - 1)) + 1
            Next
        Next
    Next

    SetVal2
End Function

Function SetVal2()
    ColGraph.Show 1
    Do While ColGraph.Visible = True
        DoEvents
    Loop
    If ColGraph.MaxVal = -1 Then Exit Function
ColShift ColGraph.MinVal, ColGraph.MaxVal, ColGraph.Val, ColGraph.Opt
End Function

Function ColShift(Min As Integer, Max As Integer, Val As Integer, Opt As Integer)
    On Error Resume Next
    Dim tVal As Integer
    LoadFile Target
    If Opt = 0 Then
        For i = 0 To YRes
            For j = 0 To XRes
                For k = 0 To 2
                    If Pic(j, i, k) >= Min And Pic(j, i, k) <= Max
                        PicTemp(j, i, k) = Pic(j, i, k) - Val
                    Else
                        PicTemp(j, i, k) = Pic(j, i, k)
                    End If
                Next
            Next
        Next
    Else
        For i = 0 To YRes
            For j = 0 To XRes
                For k = 0 To 2
                    PicTemp(j, i, k) = Pic(j, i, k)
                Next
            Next
        Next
        For i = 0 To YRes
            For j = 0 To XRes
                If Pic(j, i, Opt - 1) >= Min And Pic(j, i, Opt - 1) <= Max
                    tVal = Pic(j, i, Opt - 1) - Val
                    If tVal < 0 Then tVal = 0
                    PicTemp(j, i, Opt - 1) = tVal
                Else
                    PicTemp(j, i, Opt - 1) = Pic(j, i, Opt - 1)
                End If
            Next
        Next
    End If
    Draw
End Function

Function SavePic(Filename As String)
    On Error Resume Next
    SavePicture Target.Image, Filename
End Function

Dim n1, n2, m1, m2 As Integer
Dim x(400, 400) As Integer
Dim xr(400, 400), xi(400, 400) As Integer

Private Sub Command1_Click()
    n1 = 0
    For i = 1 To Picture1.ScaleWidth Step 15
        n1 = n1 + 1
        n2 = 0
        ' code
End Sub
For j = 1 To Picture1.ScaleWidth Step 15
    warna = Picture1.Point(i, j)
    r = Int((warna And RGB(255, 0, 0)) / 265)
    g = Int(Int((warna And RGB(0, 255, 0)) / 256) / 256)
    b = Int((warna And RGB(0, 0, 255)) / 256)
    n2 = n2 + 1
    x(n1, n2) = Int((r + g + b) / 3)
    Picture1.PSet (i, j), RGB(x(n1, n2), x(n1, n2), x(n1, n2))
Next j
Next i
Picture2.ScaleHeight = m1 + 1
Picture2.ScaleWidth = m2 + 1
Picture3.ScaleHeight = m1 + 1
Picture3.ScaleWidth = m2 + 1
For i = 1 To m1
    For j = 1 To m2
        fr = 0
        fi = 0
        For k1 = 1 To n1
            For k2 = 1 To n2
                fr = fr + x(k1, k2) * Cos(6.28 * (i * k1 / m1 + j * k2 / m2))
                fi = fi - x(k1, k2) * Sin(6.28 * (i * k1 / m1 + j * k2 / m2))
            Next k2
        Next k1
        w = 255 * Abs(fr) / (n1 * n2)
        Picture2.Line (i - 0.5, j - 0.5)-(i + 0.5, j + 0.5), RGB(w, w, w), BF
        w = 255 * Abs(fi) / (n1 * n2)
        Picture3.Line (i - 0.5, j - 0.5)-(i + 0.5, j + 0.5), RGB(w, w, w), BF
        xr(i, j) = fr
        xi(i, j) = fi
    Next j
Next i
End Sub

Private Sub Command2_Click()
    Dim xa(100, 100), xg(100, 100) As Integer
    Picture4.ScaleHeight = m1 + 1
    Picture4.ScaleWidth = m2 + 1
    Picture5.ScaleHeight = m1 + 1
    Picture5.ScaleWidth = m2 + 1
    xam = 0
    xgm = 0
    For i = 1 To m1
        For j = 1 To m2
            xa(i, j) = (xr(i, j) ^ 2 + xi(i, j) ^ 2) ^ 0.5
            xg(i, j) = xi(i, j) / xr(i, j)
            If xa(i, j) > xam Then xam = xa(i, j)
            If Abs(xg(i, j)) > xgm Then xgm = Abs(xg(i, j))
        Next j
    Next i
    For i = 1 To m1
        For j = 1 To m2
            w = Int(256 * xa(i, j) / xam)
            Picture4.Line (i - 0.5, j - 0.5)-(i + 0.5, j + 0.5), RGB(w, w, w), BF
        Next j
    Next i
End Sub
' $w = \text{Int}(256 * \text{Abs}(xg(i, j)) / xgm)$
Picture5.Line (i - 0.5, j - 0.5)-(i + 0.5, j + 0.5),
RGB(w, w, w), BF
    Next j
    Next i
End Sub

Private Sub Command3_Click()
Unload Me
End Sub

Private Sub Form_Load()
m1 = 16: m2 = 16
End Sub