

## Apéndice A

El código programado en Visual Studio 2005, fue el siguiente:

```
Imports Microsoft.Office.Core
Imports Microsoft.Office.Interop
Public Class MainForm
    Dim BlockTable(2500, 58) As Double
    Dim MaxRows As Integer
    Dim MaxCols As Integer

    Private Sub MainForm_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        gbTelTpC.Hide()
        gbGTelT.Hide()
        gbmodMMc.Hide()
        gbmodMM1K.Hide()
        gbmodMM1.Hide()
        MaxRows = 2500
        MaxCols = 58
    End Sub

    Private Function GetFileName(ByVal BlockType As String) As String
        Dim FileName As String
        FileName = ""
        If BlockType = "Erlang B" Then
            FileName = "erlangb.txt"
        ElseIf BlockType = "Extended Erlang B 10%" Then
            FileName = "exterlangb10.txt"
        ElseIf BlockType = "Extended Erlang B 20%" Then
            FileName = "exterlangb20.txt"
        ElseIf BlockType = "Extended Erlang B 30%" Then
            FileName = "exterlangb30.txt"
        ElseIf BlockType = "Extended Erlang B 40%" Then
            FileName = "exterlangb40.txt"
        ElseIf BlockType = "Extended Erlang B 50%" Then
            FileName = "exterlangb50.txt"
        ElseIf BlockType = "Extended Erlang B 60%" Then
            FileName = "exterlangb60.txt"
        ElseIf BlockType = "Extended Erlang B 70%" Then
            FileName = "exterlangb70.txt"
        ElseIf BlockType = "Extended Erlang B 80%" Then
            FileName = "exterlangb80.txt"
        ElseIf BlockType = "Extended Erlang B 90%" Then
            FileName = "exterlangb90.txt"
        ElseIf BlockType = "Erlang C" Then
            FileName = "erlangc.txt"
        ElseIf BlockType = "Poisson" Then
            FileName = "poisson.txt"
        End If
        GetFileName = FileName
    End Function

    Private Function UpdateMatrix(ByVal FileName As String) As
Boolean
        If FileName <> "" Then
            Using MyReader As New _
                Microsoft.VisualBasic.FileIO.TextFieldParser(FileName)
                MyReader.TextFieldType = FileIO.FieldType.Delimited
                MyReader.SetDelimiters(vbTab)
            End Using
        End If
    End Function
End Class
```

```

        Dim currentRow As String()
        Dim i As Integer
        Dim j As Integer
        Dim f As Boolean
        i = 0
        j = 0
        f = True
        While Not MyReader.EndOfData And f
            Try
                currentRow = MyReader.ReadFields()
                Dim currentField As String
                For Each currentField In currentRow
                    If Double.TryParse(currentField,
BlockTable(i, j)) Then
                        If j >= MaxCols Then
                            i = i + 1
                            j = 0
                        End If
                        If i >= MaxRows Then
                            f = False
                            UpdateMatrix = False
                        End If
                        j = j + 1
                    End If
                Next
            Catch ex As
                Microsoft.VisualBasic.FileIO.MalformedLineException
                MsgBox("Line " & ex.Message & _
                    "is not valid and will be skipped.")
                UpdateMatrix = False
            End Try
        End While
    End Using
End If
UpdateMatrix = True
End Function
Private Sub btnCalculate2_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCalculate2.Click
    Dim FileName As String
    If cbBlockingFormula2.SelectedItem <> Nothing Then
        FileName =
GetFileName(cbBlockingFormula2.SelectedItem.ToString)
        UpdateMatrix(FileName)
        Dim NoC As Double
        Dim ACHT As Double
        Dim OfferedLoad As Double
        Dim Trunks As Integer
        Dim j As Integer
        NoC = 0.0
        ACHT = 0.0
        Trunks = -1
        If cbGOS2.SelectedIndex >= 0 Then
            j = cbGOS2.SelectedIndex
            If Double.TryParse(tbNoC.Text, NoC) Then
                If Double.TryParse(tbACHT.Text, ACHT) Then
                    OfferedLoad = (ACHT / 3600) * NoC
                    tbOfferedLoad2.Text =
FormatNumber(OfferedLoad, 4, vbFalse)
                    For Trunks = 0 To MaxRows
                        If BlockTable(Trunks, j) >= OfferedLoad
Then

```

```

                Exit For
            End If
        Next
    Else
        MsgBox("Error: Please select an Average Call
Holding Time acceptable!")
    End If
    Else
        MsgBox("Error: Please select a number of calls
acceptable!")
    End If
    Else
        MsgBox("Please select a grade of service!")
    End If
    If Trunks >= 0 And Trunks < 2500 Then
        Trunks = Trunks + 1
        tbNoT2.Text = Trunks.ToString
    Else
        If Trunks >= 2500 Then
            MsgBox("More than 2500 Trunks will be assigned!")
        End If
    End If
    Else
        MsgBox("Please select a Blocking Formula!")
    End If
End Sub

Private Sub btnCalculate_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCalculate.Click
    Dim FileName As String
    If cbBlockingFormula.SelectedItem <> Nothing Then
        FileName =
GetFileName(cbBlockingFormula.SelectedItem.ToString)
        UpdateMatrix(FileName)
        Dim Subscribers As Double
        Dim TrafficPerSub As Double
        Dim OfferedLoad As Double
        Dim Trunks As Integer
        Dim j As Integer
        Subscribers = 0.0
        TrafficPerSub = 0.0
        OfferedLoad = 0.0
        Trunks = -1
        If cbGOS.SelectedIndex >= 0 Then
            j = cbGOS.SelectedIndex
            If Double.TryParse(tbSubscribers.Text, Subscribers)
Then
                If Double.TryParse(tbTraffic.Text, TrafficPerSub)
Then
                    OfferedLoad = Subscribers * TrafficPerSub
                    tbOfferedLoad.Text =
FormatNumber(OfferedLoad, 4, vbFalse)
                    For Trunks = 0 To MaxRows
                        If BlockTable(Trunks, j) >= OfferedLoad
Then
                            Exit For
                        End If
                    Next
                Else
                    MsgBox("Error: Please select a Traffic per
subscriber acceptable!")
                End If
            End If
        End If
    End If
End Sub

```

```

        End If
    Else
        MsgBox("Error: Please select a number of
Subscribers acceptable!")
    End If
    If Trunks >= 0 And Trunks < 2500 Then
        Trunks = Trunks + 1
        tbNoT.Text = Trunks.ToString
    Else
        If Trunks >= 2500 Then
            MsgBox("More than 2500 Trunks will be
assigned!")
        End If
    End If
Else
    MsgBox("Please select a grade of service!")
End If
Else
    MsgBox("Please select a Blocking Formula!")
End If
End Sub

Private Sub btnCalculate3_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCalculate3.Click
    Dim LenghtOfSystem As Double
    Dim LenghtOfQueue As Double
    Dim WaitingTimeSystem As Double
    Dim WaitingTimeQueue As Double
    Dim TrafficIntensity As Double
    Dim Probability As Double
    Dim ArrivalRate As Double
    Dim ServiceRate As Double
    Dim Packets As Double

    LenghtOfSystem = 0.0
    LenghtOfQueue = 0.0
    WaitingTimeSystem = 0.0
    WaitingTimeQueue = 0.0
    TrafficIntensity = 0.0
    Probability = 0.0
    ArrivalRate = tbar1.Text
    ServiceRate = tbsr1.Text
    Packets = tbppc1.Text

    If ArrivalRate >= ServiceRate Then
        MsgBox("Error: Not Stable System, Overflow! Please select
a minor arrival rate")
    Else
        If ServiceRate = 0 Then
            MsgBox("Error: Infinite with zero!, please give
another number")
        Else
            TrafficIntensity = ArrivalRate / ServiceRate
            tbtil.Text = FormatNumber(TrafficIntensity, 4,
vbFalse)
            LenghtOfSystem = TrafficIntensity / (1 -
TrafficIntensity)
            tbmls1.Text = FormatNumber(LenghtOfSystem, 4,
vbFalse)

```

```

        LenghtOfQueue = TrafficIntensity ^ 2 / (1 -
TrafficIntensity)
        tbmloQ1.Text = FormatNumber(LenghtOfQueue, 4,
vbFalse)
        WaitingTimeSystem = 1 / (ServiceRate - ArrivalRate)
        tbmwtiS1.Text = FormatNumber(WaitingTimeSystem, 4,
vbFalse)
        WaitingTimeQueue = TrafficIntensity / (ServiceRate -
ArrivalRate)
        tbmwtiQ1.Text = FormatNumber(WaitingTimeQueue, 4,
vbFalse)
        Probability = TrafficIntensity ^ Packets * (1 -
TrafficIntensity)
        tbpNpe1.Text = FormatNumber(Probability, 4, vbFalse)

        End If
    End If

End Sub

Private Sub tbsr1_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tbsr1.LostFocus
    If tbsr1.Text = "" Then
        Me.btnCalculate3.Enabled = False
    End If
End Sub

Private Sub tbsr1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles tbsr1.TextChanged
    If tbsr1.Text <> "" And tbar1.Text <> "" And tbppc1.Text <>
"" Then
        Me.btnCalculate3.Enabled = True
    End If
End Sub

Private Sub tbar1_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tbar1.LostFocus
    If tbar1.Text = "" Then
        Me.btnCalculate3.Enabled = False
    End If
End Sub

Private Sub tbar1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles tbar1.TextChanged
    If tbsr1.Text <> "" And tbar1.Text <> "" And tbppc1.Text <>
"" Then
        Me.btnCalculate3.Enabled = True
    End If
End Sub

Private Sub tbppc1_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tbppc1.LostFocus
    If tbppc1.Text = "" Then
        Me.btnCalculate3.Enabled = False
    End If
End Sub

Private Sub tbppc1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles tbppc1.TextChanged

```

```

        If tbsr1.Text <> "" And tbar1.Text <> "" And tbppc1.Text <>
"" Then
            Me.btnCalculate3.Enabled = True
        End If
    End Sub

    Private Sub btnCalculate4_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCalculate4.Click
        Dim LenghtOfSystem2 As Double
        Dim LenghtOfQueue2 As Double
        Dim WaitingTimeSystem2 As Double
        Dim WaitingTimeQueue2 As Double
        Dim TrafficIntensity2 As Double
        Dim Probability2 As Double
        Dim ArrivalRate2 As Double
        Dim ServiceRate2 As Double
        Dim Packets2 As Double
        Dim K As Double
        Dim PK As Double
        Dim EffectiveAR2 As Double

        LenghtOfSystem2 = 0.0
        LenghtOfQueue2 = 0.0
        WaitingTimeSystem2 = 0.0
        WaitingTimeQueue2 = 0.0
        TrafficIntensity2 = 0.0
        Probability2 = 0.0
        PK = 0.0
        EffectiveAR2 = 0.0
        ArrivalRate2 = tbar2.Text
        ServiceRate2 = tbsr2.Text
        Packets2 = tbppc2.Text
        K = tbtnop2.Text
        TrafficIntensity2 = ArrivalRate2 / ServiceRate2

        If ServiceRate2 = 0 Or ArrivalRate2 = 0 Then
            MsgBox("Error cannot divide by zero!")

        Else
            If TrafficIntensity2 = 1 Then
                If Packets2 >= K Then
                    MsgBox("Error: Packets Must be minor than the
Total Number of Packets")

                Else
                    vbFalse)          tbt12.Text = FormatNumber(TrafficIntensity2, 4,
                    LenghtOfSystem2 = K / 2
                    vbFalse)          tbtmlos2.Text = FormatNumber(LenghtOfSystem2, 4,
                    PK = 1 / (K + 1)
                    EffectiveAR2 = ArrivalRate2 * (1 - PK)
                    vbFalse)          tbear2.Text = FormatNumber(EffectiveAR2, 4,
                    WaitingTimeSystem2 = LenghtOfSystem2 /
                    EffectiveAR2
                    tbtmwtiS2.Text = FormatNumber(WaitingTimeSystem2,
                    4, vbFalse)
                    WaitingTimeQueue2 = WaitingTimeSystem2 - (1 /
                    ServiceRate2)

```

```

        tbmwtiQ2.Text = FormatNumber(WaitingTimeQueue2,
4, vbFalse)
        LenghtOfQueue2 = EffectiveAR2 * WaitingTimeQueue2
        tbmloQ2.Text = FormatNumber(LenghtOfQueue2, 4,
vbFalse)
        Probability2 = 1 / (K + 1)
        tbpNpe2.Text = FormatNumber(Probability2, 4,
vbFalse)

        End If

    Else
        If Packets2 >= K Then
            MsgBox("Error: Packets Must be minor than the
Total Number of Packets")

        Else
            If K = 0 Or K = -1 Then
                MsgBox("Error: Ecuation undefined, please
select other Total Number of Packets")

            Else
                tbtI2.Text = FormatNumber(TrafficIntensity2,
4, vbFalse)
                LenghtOfSystem2 = ((TrafficIntensity2 / (1 -
TrafficIntensity2)) - (((K + 1) * (TrafficIntensity2 ^ (K + 1))) / (1
- (TrafficIntensity2 ^ (K + 1)))))
                tbmloS2.Text = FormatNumber(LenghtOfSystem2,
4, vbFalse)
                PK = (((TrafficIntensity2 ^ K) * (1 -
TrafficIntensity2)) / (1 - (TrafficIntensity2 ^ (K + 1))))
                EffectiveAR2 = ArrivalRate2 * (1 - PK)
                tbear2.Text = FormatNumber(EffectiveAR2, 4,
vbFalse)
                WaitingTimeSystem2 = LenghtOfSystem2 /
EffectiveAR2
                tbmwtiS2.Text =
FormatNumber(WaitingTimeSystem2, 4, vbFalse)
                WaitingTimeQueue2 = WaitingTimeSystem2 - (1 /
ServiceRate2)
                tbmwtiQ2.Text =
FormatNumber(WaitingTimeQueue2, 4, vbFalse)
                LenghtOfQueue2 = EffectiveAR2 *
WaitingTimeQueue2
                tbmloQ2.Text = FormatNumber(LenghtOfQueue2,
4, vbFalse)
                Probability2 = (((TrafficIntensity2 ^
Packets2) * (1 - TrafficIntensity2)) / (1 - (TrafficIntensity2 ^ (K +
1))))
                tbpNpe2.Text = FormatNumber(Probability2, 4,
vbFalse)

                End If
            End If
        End If
    End If

End Sub

```

```

Private Sub tbsr2_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tbsr2.LostFocus
    If tbsr2.Text = "" Then
        Me.btnCalculate4.Enabled = False
    End If
End Sub

Private Sub tbsr2_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles tbsr2.TextChanged
    If tbsr2.Text <> "" And tbar2.Text <> "" And tbppc2.Text <>
"" And tbtnop2.Text <> "" Then
        Me.btnCalculate4.Enabled = True
    End If
End Sub

Private Sub tbar2_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tbar2.LostFocus
    If tbar2.Text = "" Then
        Me.btnCalculate4.Enabled = False
    End If
End Sub

Private Sub tbar2_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles tbar2.TextChanged
    If tbsr2.Text <> "" And tbar2.Text <> "" And tbppc2.Text <>
"" And tbtnop2.Text <> "" Then
        Me.btnCalculate4.Enabled = True
    End If
End Sub

Private Sub tbtnop2_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tbtnop2.LostFocus
    If tbtnop2.Text = "" Then
        Me.btnCalculate4.Enabled = False
    End If
End Sub

Private Sub tbtnop2_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles tbtnop2.TextChanged
    If tbsr2.Text <> "" And tbar2.Text <> "" And tbppc2.Text <>
"" And tbtnop2.Text <> "" Then
        Me.btnCalculate4.Enabled = True
    End If
End Sub

Private Sub tbppc2_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tbppc2.LostFocus
    If tbppc2.Text = "" Then
        Me.btnCalculate4.Enabled = False
    End If
End Sub

Private Sub tbppc2_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles tbppc2.TextChanged
    If tbsr2.Text <> "" And tbar2.Text <> "" And tbppc2.Text <>
"" And tbtnop2.Text <> "" Then
        Me.btnCalculate4.Enabled = True
    End If
End Sub

```



```

Private Sub btnCalculate5_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCalculate5.Click
    Dim LenghtOfSystem3 As Double
    Dim LenghtOfQueue3 As Double
    Dim WaitingTimeSystem3 As Double
    Dim WaitingTimeQueue3 As Double
    Dim Probability3 As Double
    Dim ArrivalRate3 As Double
    Dim ServiceRate3 As Double
    Dim TI3 As Double
    Dim C As Double
    Dim Po As Double
    Dim S As Double

    LenghtOfSystem3 = 0.0
    LenghtOfQueue3 = 0.0
    WaitingTimeSystem3 = 0.0
    WaitingTimeQueue3 = 0.0
    TI3 = 0.0
    Probability3 = 0.0
    Po = 0.0
    S = 0.0
    ArrivalRate3 = tbar3.Text
    ServiceRate3 = tbsr3.Text

    If cbs3.SelectedItem = Nothing Then
        MsgBox("Please select a Number of Servers")
    Else
        C = cbs3.SelectedIndex + 1
        TI3 = ArrivalRate3 / (ServiceRate3 * C)
        If C = 1 Then
            If ArrivalRate3 >= (ServiceRate3 * C) Then
                MsgBox("Error: Not Stable System, Overflow!
Please select a minor arrival rate")
            Else
                If ServiceRate3 = 0 Or ArrivalRate3 = 0 Then
                    MsgBox("Error: Infinite with zero!, please
give another number")
                Else
                    tbt3.Text = FormatNumber(TI3, 4, vbFalse)
                    Po = 1 - TI3
                    tbp0.Text = FormatNumber(Po, 4, vbFalse)
                    LenghtOfQueue3 = ((TI3 ^ 2) / (1 - TI3))
                    tbmloQ3.Text = FormatNumber(LenghtOfQueue3,
4, vbFalse)
                    WaitingTimeQueue3 = (LenghtOfQueue3 /
ArrivalRate3)
                    tbmwtiQ3.Text =
FormatNumber(WaitingTimeQueue3, 4, vbFalse)
                    WaitingTimeSystem3 = (WaitingTimeQueue3 + (1
/ ServiceRate3))
                    tbmwtiS3.Text =
FormatNumber(WaitingTimeSystem3, 4, vbFalse)
                    LenghtOfSystem3 = (ArrivalRate3 *
WaitingTimeSystem3)
                    tbmlos3.Text = FormatNumber(LenghtOfSystem3,
4, vbFalse)
                End If
            End If
        End If
    End If
End Sub

```

```

        S = (C * TI3)
        tbbs3.Text = FormatNumber(S, 4, vbFalse)
    End If
End If
End If

If C = 2 Then
    If ArrivalRate3 >= (ServiceRate3 * C) Then
        MsgBox("Error: Not Stable System, Overflow!
Please select a minor arrival rate")
    Else
        If ServiceRate3 = 0 Or ArrivalRate3 = 0 Then
            MsgBox("Error: Infinite with zero!, please
give another number")
        Else
            tbti3.Text = FormatNumber(TI3, 4, vbFalse)
            Po = (1 - TI3) / (1 + TI3)
            tbp0.Text = FormatNumber(Po, 4, vbFalse)
            LenghtOfQueue3 = ((2 * (TI3 ^ 3)) / (1 - (TI3
^ 2)))
            tbmloQ3.Text = FormatNumber(LenghtOfQueue3,
4, vbFalse)
            WaitingTimeQueue3 = (LenghtOfQueue3 /
ArrivalRate3)
            tbmwtiQ3.Text =
FormatNumber(WaitingTimeQueue3, 4, vbFalse)
            WaitingTimeSystem3 = (WaitingTimeQueue3 + (1
/ ServiceRate3))
            tbmwtiS3.Text =
FormatNumber(WaitingTimeSystem3, 4, vbFalse)
            LenghtOfSystem3 = (ArrivalRate3 *
WaitingTimeSystem3)
            tbmlos3.Text = FormatNumber(LenghtOfSystem3,
4, vbFalse)
            S = (C * TI3)
            tbbs3.Text = FormatNumber(S, 4, vbFalse)
        End If
    End If
End If

If C = 3 Then
    If ArrivalRate3 >= (ServiceRate3 * C) Then
        MsgBox("Error: Not Stable System, Overflow!
Please select a minor arrival rate")
    Else
        If ServiceRate3 = 0 Or ArrivalRate3 = 0 Then
            MsgBox("Error: Infinite with zero!, please
give another number")
        Else
            tbti3.Text = FormatNumber(TI3, 4, vbFalse)
            Po = ((2 * (1 - TI3)) / (2 + (4 * TI3) + (3 *
(TI3 ^ 2))))
            tbp0.Text = FormatNumber(Po, 4, vbFalse)
            LenghtOfQueue3 = ((9 * (TI3 ^ 4)) / (2 + (2 *
TI3) - (TI3 ^ 2) - (3 * (TI3 ^ 3))))
            tbmloQ3.Text = FormatNumber(LenghtOfQueue3,
4, vbFalse)

```

```

WaitingTimeQueue3 = (LenghtOfQueue3 /
ArrivalRate3)
tbnwtiQ3.Text =
FormatNumber(WaitingTimeQueue3, 4, vbFalse)
WaitingTimeSystem3 = (WaitingTimeQueue3 + (1
/ ServiceRate3))
tbnwtiS3.Text =
FormatNumber(WaitingTimeSystem3, 4, vbFalse)
LenghtOfSystem3 = (ArrivalRate3 *
WaitingTimeSystem3)
tbnlos3.Text = FormatNumber(LenghtOfSystem3,
4, vbFalse)
S = (C * TI3)
tbbs3.Text = FormatNumber(S, 4, vbFalse)
End If
End If
End If

If C = 4 Then
If ArrivalRate3 >= (ServiceRate3 * C) Then
MsgBox("Error: Not Stable System, Overflow!
Please select a minor arrival rate")

Else
If ServiceRate3 = 0 Or ArrivalRate3 = 0 Then
MsgBox("Error: Infinite with zero!, please
give another number")

Else
tbtI3.Text = FormatNumber(TI3, 4, vbFalse)
Po = (((3 * (1 - TI3)) / (3 + (9 * TI3) + (12
* (TI3 ^ 2)) + (8 * (TI3 ^ 3))))
tbp0.Text = FormatNumber(Po, 4, vbFalse)
LenghtOfQueue3 = (((32 * (TI3 ^ 5)) / (3 + (6
* TI3) + (3 + (TI3 ^ 2)) - (4 * (TI3 ^ 3)) - (8 * (TI3 ^ 4))))
tbnloQ3.Text = FormatNumber(LenghtOfQueue3,
4, vbFalse)

WaitingTimeQueue3 = (LenghtOfQueue3 /
ArrivalRate3)
tbnwtiQ3.Text =
FormatNumber(WaitingTimeQueue3, 4, vbFalse)
WaitingTimeSystem3 = (WaitingTimeQueue3 + (1
/ ServiceRate3))
tbnwtiS3.Text =
FormatNumber(WaitingTimeSystem3, 4, vbFalse)
LenghtOfSystem3 = (ArrivalRate3 *
WaitingTimeSystem3)
tbnlos3.Text = FormatNumber(LenghtOfSystem3,
4, vbFalse)

S = (C * TI3)
tbbs3.Text = FormatNumber(S, 4, vbFalse)
End If
End If
End If

If C = 5 Then
If ArrivalRate3 >= (ServiceRate3 * C) Then
MsgBox("Error: Not Stable System, Overflow!
Please select a minor arrival rate")

Else

```

```

        If ServiceRate3 = 0 Or ArrivalRate3 = 0 Then
            MsgBox("Error: Infinite with zero!, please
give another number")

        Else
            tbtI3.Text = FormatNumber(TI3, 4, vbFalse)
            Po = ((24 * (1 - TI3)) / (24 + (96 * TI3) +
(180 * (TI3 ^ 2)) + (200 * (TI3 ^ 3)) + (124 * (TI3 ^ 4))))
            tbp0.Text = FormatNumber(Po, 4, vbFalse)
            LenghtOfQueue3 = ((625 * (TI3 ^ 6)) / (24 +
(72 * TI3) + (84 * (TI3 ^ 2)) + (20 * (TI3 ^ 3)) - (75 * (TI3 ^ 4)) -
(125 * (TI3 ^ 5))))
            tbmloQ3.Text = FormatNumber(LenghtOfQueue3,
4, vbFalse)
            WaitingTimeQueue3 = (LenghtOfQueue3 /
ArrivalRate3)
            tbmwtiQ3.Text =
FormatNumber(WaitingTimeQueue3, 4, vbFalse)
            WaitingTimeSystem3 = (WaitingTimeQueue3 + (1
/ ServiceRate3))
            tbmwtiS3.Text =
FormatNumber(WaitingTimeSystem3, 4, vbFalse)
            LenghtOfSystem3 = (ArrivalRate3 *
WaitingTimeSystem3)
            tbmlos3.Text = FormatNumber(LenghtOfSystem3,
4, vbFalse)
            S = (C * TI3)
            tbbs3.Text = FormatNumber(S, 4, vbFalse)
        End If
    End If
End If

If C = 6 Then
    If ArrivalRate3 >= (ServiceRate3 * C) Then
        MsgBox("Error: Not Stable System, Overflow!
Please select a minor arrival rate")

    Else
        If ServiceRate3 = 0 Or ArrivalRate3 = 0 Then
            MsgBox("Error: Infinite with zero!, please
give another number")

        Else
            tbtI3.Text = FormatNumber(TI3, 4, vbFalse)
            Po = ((5 * (1 - TI3)) / (5 + (25 * TI3) + (60
* (TI3 ^ 2)) + (90 * (TI3 ^ 3)) + (90 * (TI3 ^ 4)) + (54 * (TI3 ^
5))))
            tbp0.Text = FormatNumber(Po, 4, vbFalse)
            LenghtOfQueue3 = ((324 * (TI3 ^ 7)) / (5 +
(20 * TI3) + (35 * (TI3 ^ 2)) + (30 * (TI3 ^ 3)) - (36 * (TI3 ^ 5)) -
(54 * (TI3 ^ 6))))
            tbmloQ3.Text = FormatNumber(LenghtOfQueue3,
4, vbFalse)
            WaitingTimeQueue3 = (LenghtOfQueue3 /
ArrivalRate3)
            tbmwtiQ3.Text =
FormatNumber(WaitingTimeQueue3, 4, vbFalse)
            WaitingTimeSystem3 = (WaitingTimeQueue3 + (1
/ ServiceRate3))
            tbmwtiS3.Text =
FormatNumber(WaitingTimeSystem3, 4, vbFalse)

```

```

WaitingTimeSystem3)          LenghtOfSystem3 = (ArrivalRate3 *
                              tbmlos3.Text = FormatNumber(LenghtOfSystem3,
4, vbFalse)                  S = (C * TI3)
                              tbbs3.Text = FormatNumber(S, 4, vbFalse)
                              End If
                              End If
                              End If
                              End If

End Sub

Private Sub ExitToolStripMenuItem1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ExitToolStripMenuItem1.Click
    Application.Exit()
End Sub

Private Sub QueingToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
QueingToolStripMenuItem.Click
    gbTelTpC.Hide()
    gbGTelT.Hide()
    gbmodMMc.Show()
    gbmodMM1K.Show()
    gbmodMM1.Show()
End Sub

Private Sub AnalyzeToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
AnalyzeToolStripMenuItem.Click
    gbTelTpC.Show()
    gbGTelT.Show()
    gbmodMMc.Hide()
    gbmodMM1K.Hide()
    gbmodMM1.Hide()
End Sub

Private Sub tbsr3_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tbsr3.LostFocus
    If tbsr3.Text = "" Then
        Me.btnCalculate5.Enabled = False
    End If
End Sub

Private Sub tbsr3_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles tbsr3.TextChanged
    If tbsr3.Text <> "" And tbar3.Text <> "" Then
        Me.btnCalculate5.Enabled = True
    End If
End Sub

Private Sub tbar3_LostFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tbar3.LostFocus
    If tbar3.Text = "" Then
        Me.btnCalculate5.Enabled = False
    End If
End Sub

```

```

Private Sub tbar3_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles tbar3.TextChanged
    If tbar3.Text <> "" And tbar3.Text <> "" Then
        Me.btnCalculate5.Enabled = True
    End If
End Sub

Private Sub ContentsToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ContentsToolStripMenuItem.Click

    Dim pa As String, pat1 As String, pat2 As String

    pat1 = "C:\Program Files\Adobe\Acrobat
7.0\Acrobat\Acrobat.exe "
    pat2 = "C:\TRAFFIC ANALYZER\Telephones.pdf"
    pa = pat1 & pat2

    Shell(pa, vbNormalFocus)

End Sub

Private Sub IndexToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
IndexToolStripMenuItem.Click
    Dim pa As String, pat1 As String, pat2 As String

    pat1 = "C:\Program Files\Adobe\Acrobat
7.0\Acrobat\Acrobat.exe "
    pat2 = "C:\TRAFFIC ANALYZER\QueueingS.pdf"
    pa = pat1 & pat2

    Shell(pa, vbNormalFocus)
End Sub

End Class

```