Sub Command0\_Click()

MsgBox "Select the Filename"

Dim OpenFile As OPENFILENAME

Dim lReturn As Long

Dim sFilter As String

OpenFile.lStructSize = Len(OpenFile)

OpenFile.hwndOwner = Me.Hwnd

OpenFile.hInstance = -1

sFilter = "Excel Files (\*.xlsx)" & Chr(0) & "\*.xlsx" & Chr(0)

OpenFile.lpstrFilter = sFilter

OpenFile.nFilterIndex = 1

OpenFile.lpstrFile = String(257, 0)

OpenFile.nMaxFile = Len(OpenFile.lpstrFile) - 1

OpenFile.lpstrFileTitle = OpenFile.lpstrFile

OpenFile.nMaxFileTitle = OpenFile.nMaxFile

OpenFile.lpstrInitialDir = "C:\"

OpenFile.lpstrTitle = "Use the Comdlg API not the OCX"

OpenFile.flags = 0

lReturn = GetOpenFileName(OpenFile)

If lReturn = 0 Then

' MsgBox "The User pressed the Cancel Button"

'nothing

Else

'MsgBox "The user Chose " &

fname = Trim(OpenFile.lpstrFile)

MsgBox fname

End If

For i = Len(fname) To 1 Step -1

If Mid(fname, i, 1) = "\" Then

folderfrompath = Left(fname, i)

Exit For

End If

Next

Dim testSQL As String

Dim qd As DAO.QueryDef

strPath = folderfrompath

Set dbs = CurrentDb

'clear out existing data

dbs.Execute "Delete \* From spare", dbFailOnError

DoCmd.SetWarnings False

'DoCmd.TransferSpreadsheet acImport, "IMPORT\_BIDNUMBERS", "BIDTAB", fname, False, ""

DoCmd.TransferSpreadsheet acImport, acSpreadsheetTypeExcel9, "spare", fname, True

MsgBox "Imported Successfully"

'Me.BIDNUMBERSWITHBASEPRICE.Requery

strPath1 = strPath

strPath = strPath & "vanityout"

DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_vanity\_with\_category", strPath, True

strPath = strPath1 & "notvanity.csv"

'DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_not\_vanity", strPath, True

'Me.NEWTAB.Requery

testSQL = "SELECT \* FROM spare\_not\_vanity"

'dbs.QueryDefs.Delete "tmpExport"

Set qd = dbs.CreateQueryDef("tmpExport", testSQL)

DoCmd.TransferText acExportDelim, , "tmpExport", strPath

dbs.QueryDefs.Delete "tmpExport"

MsgBox "Completed.Pls Check vanityout and notvanity files"

DoCmd.SetWarnings True

End Sub

Sub Command0\_Click()

MsgBox "Select the Filename"

Dim OpenFile As OPENFILENAME

Dim lReturn As Long

Dim sFilter As String

OpenFile.lStructSize = Len(OpenFile)

OpenFile.hwndOwner = Me.Hwnd

OpenFile.hInstance = -1

sFilter = "Excel Files (\*.xlsx)" & Chr(0) & "\*.xlsx" & Chr(0)

OpenFile.lpstrFilter = sFilter

OpenFile.nFilterIndex = 1

OpenFile.lpstrFile = String(257, 0)

OpenFile.nMaxFile = Len(OpenFile.lpstrFile) - 1

OpenFile.lpstrFileTitle = OpenFile.lpstrFile

OpenFile.nMaxFileTitle = OpenFile.nMaxFile

OpenFile.lpstrInitialDir = "C:\"

OpenFile.lpstrTitle = "Use the Comdlg API not the OCX"

OpenFile.flags = 0

lReturn = GetOpenFileName(OpenFile)

If lReturn = 0 Then

' MsgBox "The User pressed the Cancel Button"

'nothing

Else

'MsgBox "The user Chose " &

fname = Trim(OpenFile.lpstrFile)

MsgBox fname

End If

For i = Len(fname) To 1 Step -1

If Mid(fname, i, 1) = "\" Then

folderfrompath = Left(fname, i)

Exit For

End If

Next

Dim testSQL As String

Dim qd As DAO.QueryDef

strPath = folderfrompath

Set dbs = CurrentDb

'clear out existing data

dbs.Execute "Delete \* From spare", dbFailOnError

DoCmd.SetWarnings False

'DoCmd.TransferSpreadsheet acImport, "IMPORT\_BIDNUMBERS", "BIDTAB", fname, False, ""

DoCmd.TransferSpreadsheet acImport, acSpreadsheetTypeExcel9, "spare", fname, True

MsgBox "Imported Successfully"

'Me.BIDNUMBERSWITHBASEPRICE.Requery

strPath1 = strPath

strPath = strPath & "vanityout"

DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_vanity\_with\_category", strPath, True

strPath = strPath1 & "notvanity.csv"

'DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_not\_vanity", strPath, True

'Me.NEWTAB.Requery

testSQL = "SELECT \* FROM spare\_not\_vanity"

'dbs.QueryDefs.Delete "tmpExport"

Set qd = dbs.CreateQueryDef("tmpExport", testSQL)

DoCmd.TransferText acExportDelim, , "tmpExport", strPath

dbs.QueryDefs.Delete "tmpExport"

MsgBox "Completed.Pls Check vanityout and notvanity files"

DoCmd.SetWarnings True

End Sub

Sub Command0\_Click()

MsgBox "Select the Filename"

Dim OpenFile As OPENFILENAME

Dim lReturn As Long

Dim sFilter As String

OpenFile.lStructSize = Len(OpenFile)

OpenFile.hwndOwner = Me.Hwnd

OpenFile.hInstance = -1

sFilter = "Excel Files (\*.xlsx)" & Chr(0) & "\*.xlsx" & Chr(0)

OpenFile.lpstrFilter = sFilter

OpenFile.nFilterIndex = 1

OpenFile.lpstrFile = String(257, 0)

OpenFile.nMaxFile = Len(OpenFile.lpstrFile) - 1

OpenFile.lpstrFileTitle = OpenFile.lpstrFile

OpenFile.nMaxFileTitle = OpenFile.nMaxFile

OpenFile.lpstrInitialDir = "C:\"

OpenFile.lpstrTitle = "Use the Comdlg API not the OCX"

OpenFile.flags = 0

lReturn = GetOpenFileName(OpenFile)

If lReturn = 0 Then

' MsgBox "The User pressed the Cancel Button"

'nothing

Else

'MsgBox "The user Chose " &

fname = Trim(OpenFile.lpstrFile)

MsgBox fname

End If

For i = Len(fname) To 1 Step -1

If Mid(fname, i, 1) = "\" Then

folderfrompath = Left(fname, i)

Exit For

End If

Next

Dim testSQL As String

Dim qd As DAO.QueryDef

strPath = folderfrompath

Set dbs = CurrentDb

'clear out existing data

dbs.Execute "Delete \* From spare", dbFailOnError

DoCmd.SetWarnings False

'DoCmd.TransferSpreadsheet acImport, "IMPORT\_BIDNUMBERS", "BIDTAB", fname, False, ""

DoCmd.TransferSpreadsheet acImport, acSpreadsheetTypeExcel9, "spare", fname, True

MsgBox "Imported Successfully"

'Me.BIDNUMBERSWITHBASEPRICE.Requery

strPath1 = strPath

strPath = strPath & "vanityout"

DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_vanity\_with\_category", strPath, True

strPath = strPath1 & "notvanity.csv"

'DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_not\_vanity", strPath, True

'Me.NEWTAB.Requery

testSQL = "SELECT \* FROM spare\_not\_vanity"

'dbs.QueryDefs.Delete "tmpExport"

Set qd = dbs.CreateQueryDef("tmpExport", testSQL)

DoCmd.TransferText acExportDelim, , "tmpExport", strPath

dbs.QueryDefs.Delete "tmpExport"

MsgBox "Completed.Pls Check vanityout and notvanity files"

DoCmd.SetWarnings True

End Sub

**MODULE 1**

**Sub Command0\_Click()**

**MsgBox "Select the Filename"**

**Dim OpenFile As OPENFILENAME**

**Dim lReturn As Long**

**Dim sFilter As String**

**OpenFile.lStructSize = Len(OpenFile)**

**OpenFile.hwndOwner = Me.Hwnd**

**OpenFile.hInstance = -1**

**sFilter = "Excel Files (\*.xlsx)" & Chr(0) & "\*.xlsx" & Chr(0)**

**OpenFile.lpstrFilter = sFilter**

**OpenFile.nFilterIndex = 1**

**OpenFile.lpstrFile = String(257, 0)**

**OpenFile.nMaxFile = Len(OpenFile.lpstrFile) - 1**

**OpenFile.lpstrFileTitle = OpenFile.lpstrFile**

**OpenFile.nMaxFileTitle = OpenFile.nMaxFile**

**OpenFile.lpstrInitialDir = "C:\"**

**OpenFile.lpstrTitle = "Use the Comdlg API not the OCX"**

**OpenFile.flags = 0**

**lReturn = GetOpenFileName(OpenFile)**

**If lReturn = 0 Then**

**' MsgBox "The User pressed the Cancel Button"**

**'nothing**

**Else**

**'MsgBox "The user Chose " &**

**fname = Trim(OpenFile.lpstrFile)**

**MsgBox fname**

**End If**

**For i = Len(fname) To 1 Step -1**

**If Mid(fname, i, 1) = "\" Then**

**folderfrompath = Left(fname, i)**

**Exit For**

**End If**

**Next**

**Dim testSQL As String**

**Dim qd As DAO.QueryDef**

**strPath = folderfrompath**

**Set dbs = CurrentDb**

**'clear out existing data**

**dbs.Execute "Delete \* From spare", dbFailOnError**

**DoCmd.SetWarnings False**

**'DoCmd.TransferSpreadsheet acImport, "IMPORT\_BIDNUMBERS", "BIDTAB", fname, False, ""**

**DoCmd.TransferSpreadsheet acImport, acSpreadsheetTypeExcel9, "spare", fname, True**

**MsgBox "Imported Successfully"**

**'Me.BIDNUMBERSWITHBASEPRICE.Requery**

**strPath1 = strPath**

**strPath = strPath & "vanityout"**

**DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_vanity\_with\_category", strPath, True**

**strPath = strPath1 & "notvanity.csv"**

**'DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_not\_vanity", strPath, True**

**'Me.NEWTAB.Requery**

**testSQL = "SELECT \* FROM spare\_not\_vanity"**

**'dbs.QueryDefs.Delete "tmpExport"**

**Set qd = dbs.CreateQueryDef("tmpExport", testSQL)**

**DoCmd.TransferText acExportDelim, , "tmpExport", strPath**

**dbs.QueryDefs.Delete "tmpExport"**

**MsgBox "Completed.Pls Check vanityout and notvanity files"**

**DoCmd.SetWarnings True**

**End Sub**

**MODULE 2**

**Sub Command0\_Click()**

**MsgBox "Select the Filename"**

**Dim OpenFile As OPENFILENAME**

**Dim lReturn As Long**

**Dim sFilter As String**

**OpenFile.lStructSize = Len(OpenFile)**

**OpenFile.hwndOwner = Me.Hwnd**

**OpenFile.hInstance = -1**

**sFilter = "Excel Files (\*.xlsx)" & Chr(0) & "\*.xlsx" & Chr(0)**

**OpenFile.lpstrFilter = sFilter**

**OpenFile.nFilterIndex = 1**

**OpenFile.lpstrFile = String(257, 0)**

**OpenFile.nMaxFile = Len(OpenFile.lpstrFile) - 1**

**OpenFile.lpstrFileTitle = OpenFile.lpstrFile**

**OpenFile.nMaxFileTitle = OpenFile.nMaxFile**

**OpenFile.lpstrInitialDir = "C:\"**

**OpenFile.lpstrTitle = "Use the Comdlg API not the OCX"**

**OpenFile.flags = 0**

**lReturn = GetOpenFileName(OpenFile)**

**If lReturn = 0 Then**

**' MsgBox "The User pressed the Cancel Button"**

**'nothing**

**Else**

**'MsgBox "The user Chose " &**

**fname = Trim(OpenFile.lpstrFile)**

**MsgBox fname**

**End If**

**For i = Len(fname) To 1 Step -1**

**If Mid(fname, i, 1) = "\" Then**

**folderfrompath = Left(fname, i)**

**Exit For**

**End If**

**Next**

**Dim testSQL As String**

**Dim qd As DAO.QueryDef**

**strPath = folderfrompath**

**Set dbs = CurrentDb**

**'clear out existing data**

**dbs.Execute "Delete \* From spare", dbFailOnError**

**DoCmd.SetWarnings False**

**'DoCmd.TransferSpreadsheet acImport, "IMPORT\_BIDNUMBERS", "BIDTAB", fname, False, ""**

**DoCmd.TransferSpreadsheet acImport, acSpreadsheetTypeExcel9, "spare", fname, True**

**MsgBox "Imported Successfully"**

**'Me.BIDNUMBERSWITHBASEPRICE.Requery**

**strPath1 = strPath**

**strPath = strPath & "vanityout"**

**DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_vanity\_with\_category", strPath, True**

**strPath = strPath1 & "notvanity.csv"**

**'DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_not\_vanity", strPath, True**

**'Me.NEWTAB.Requery**

**testSQL = "SELECT \* FROM spare\_not\_vanity"**

**'dbs.QueryDefs.Delete "tmpExport"**

**Set qd = dbs.CreateQueryDef("tmpExport", testSQL)**

**DoCmd.TransferText acExportDelim, , "tmpExport", strPath**

**dbs.QueryDefs.Delete "tmpExport"**

**MsgBox "Completed.Pls Check vanityout and notvanity files"**

**DoCmd.SetWarnings True**

**End Sub**

**MODULE 3**

**Sub Command0\_Click()**

**MsgBox "Select the Filename"**

**Dim OpenFile As OPENFILENAME**

**Dim lReturn As Long**

**Dim sFilter As String**

**OpenFile.lStructSize = Len(OpenFile)**

**OpenFile.hwndOwner = Me.Hwnd**

**OpenFile.hInstance = -1**

**sFilter = "Excel Files (\*.xlsx)" & Chr(0) & "\*.xlsx" & Chr(0)**

**OpenFile.lpstrFilter = sFilter**

**OpenFile.nFilterIndex = 1**

**OpenFile.lpstrFile = String(257, 0)**

**OpenFile.nMaxFile = Len(OpenFile.lpstrFile) - 1**

**OpenFile.lpstrFileTitle = OpenFile.lpstrFile**

**OpenFile.nMaxFileTitle = OpenFile.nMaxFile**

**OpenFile.lpstrInitialDir = "C:\"**

**OpenFile.lpstrTitle = "Use the Comdlg API not the OCX"**

**OpenFile.flags = 0**

**lReturn = GetOpenFileName(OpenFile)**

**If lReturn = 0 Then**

**' MsgBox "The User pressed the Cancel Button"**

**'nothing**

**Else**

**'MsgBox "The user Chose " &**

**fname = Trim(OpenFile.lpstrFile)**

**MsgBox fname**

**End If**

**For i = Len(fname) To 1 Step -1**

**If Mid(fname, i, 1) = "\" Then**

**folderfrompath = Left(fname, i)**

**Exit For**

**End If**

**Next**

**Dim testSQL As String**

**Dim qd As DAO.QueryDef**

**strPath = folderfrompath**

**Set dbs = CurrentDb**

**'clear out existing data**

**dbs.Execute "Delete \* From spare", dbFailOnError**

**DoCmd.SetWarnings False**

**'DoCmd.TransferSpreadsheet acImport, "IMPORT\_BIDNUMBERS", "BIDTAB", fname, False, ""**

**DoCmd.TransferSpreadsheet acImport, acSpreadsheetTypeExcel9, "spare", fname, True**

**MsgBox "Imported Successfully"**

**'Me.BIDNUMBERSWITHBASEPRICE.Requery**

**strPath1 = strPath**

**strPath = strPath & "vanityout"**

**DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_vanity\_with\_category", strPath, True**

**strPath = strPath1 & "notvanity.csv"**

**'DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel9, "spare\_not\_vanity", strPath, True**

**'Me.NEWTAB.Requery**

**testSQL = "SELECT \* FROM spare\_not\_vanity"**

**'dbs.QueryDefs.Delete "tmpExport"**

**Set qd = dbs.CreateQueryDef("tmpExport", testSQL)**

**DoCmd.TransferText acExportDelim, , "tmpExport", strPath**

**dbs.QueryDefs.Delete "tmpExport"**

**MsgBox "Completed.Pls Check vanityout and notvanity files"**

**DoCmd.SetWarnings True**

**End Sub**

**FUNCTION**

**Function chkvanity\_old(teleno) As String**

**chkvanity\_old = ""**

**pattern = Right(teleno, 5)**

**If pattern = "99999" Then**

**'MsgBox "Vanity-L1Plus"**

**chkvanity\_old = "ABCDE99999"**

**Exit Function**

**End If**

**'ABCDE77777**

**If pattern = "77777" Then**

**'MsgBox "Vanity-L1Plus"**

**chkvanity\_old = "ABCDE77777"**

**Exit Function**

**End If**

**pattern = Right(teleno, 6)**

**If chksame\_odd(pattern) = 1 Then**

**'MsgBox "Vanity-L1Plus"**

**chkvanity\_old = "ABCDXXXXXX\_ODD"**

**Exit Function**

**End If**

**'ABCDE11111**

**If Right(teleno, 5) = "11111" Then**

**'MsgBox "Vanity-L1Plus"**

**chkvanity\_old = "ABCDE11111"**

**Exit Function**

**End If**

**'ABCDXXXXXX\_EVEN**

**pattern = Right(teleno, 6)**

**If chksame\_even(pattern) = 1 Then**

**chkvanity\_old = "ABCDXXXXXX\_EVEN"**

**Exit Function**

**End If**

**'ABCDEXXXXX\_ODD**

**pattern = Right(teleno, 5)**

**If chksame\_odd(pattern) = 1 Then**

**chkvanity\_old = "ABCDEXXXXX\_ODD"**

**Exit Function**

**End If**

**'ABCDEXXXXX\_EVEN**

**pattern = Right(teleno, 5)**

**If chksame\_even(pattern) = 1 Then**

**chkvanity\_old = "ABCDEXXXXX\_EVEN"**

**Exit Function**

**End If**

**'ABCDEABCDE**

**If chkrptpttn5(teleno) = 1 Then**

**chkvanity\_old = "AXYZXYZXYZ"**

**Exit Function**

**End If**

**'AXYZXYZXYZ**

**If chkrptpttn3(teleno) = 1 Then**

**chkvanity\_old = "AXYZXYZXYZ"**

**Exit Function**

**End If**

**'ABCDUVWXYZ**

**pattern = Right(teleno, 6)**

**If chkadjacent\_old(pattern) = 1 Then**

**chkvanity\_old = "ABCDUVWXYZ"**

**Exit Function**

**End If**

**'ABCDEF9999**

**pattern = Right(teleno, 4)**

**If pattern = "9999" Then**

**chkvanity\_old = "ABCDEF9999"**

**Exit Function**

**End If**

**'ABCDEF1111**

**pattern = Right(teleno, 4)**

**If pattern = "1111" Then**

**'MsgBox "Vanity-L1Plus"**

**chkvanity\_old = "ABCDEF1111"**

**Exit Function**

**End If**

**'ABCDEVWXYZ**

**pattern = Right(teleno, 5)**

**If chkadjacent\_old(pattern) = 1 Then**

**chkvanity\_old = "ABCDEVWXYZ"**

**Exit Function**

**End If**

**'ABCDXYXYXY**

**If chkrptpttn2(teleno) = 1 Then**

**chkvanity\_old = "ABCDXYXYXY"**

**Exit Function**

**End If**

**'ABCDEFXXXX\_ODD**

**pattern = Right(teleno, 4)**

**If chksame\_odd(pattern) = 1 Then**

**chkvanity\_old = "ABCDEFXXXX\_ODD"**

**Exit Function**

**End If**

**'ABWXYZWXYZ**

**If chkrptpttn4(teleno) = 1 Then**

**chkvanity\_old = "ABWXYZWXYZ"**

**Exit Function**

**End If**

**'ABCDWXWYWZ**

**If chkrptptnadj2(teleno) = 1 Then**

**chkvanity\_old = "ABCDWXWYWZ"**

**Exit Function**

**End If**

**'ABCDXXXYYY**

**'MsgBox**

**If (chksame1(Mid(teleno, 5, 3)) = 1) And (chksame1(Mid(teleno, 8, 3)) = 1) Then**

**chkvanity\_old = "ABCDXXXYYY"**

**Exit Function**

**End If**

**'ABCDEFXXXX\_EVEN**

**pattern = Right(teleno, 4)**

**If chksame\_even(pattern) = 1 Then**

**chkvanity\_old = "ABCDEFXXXX\_EVEN"**

**Exit Function**

**End If**

**'ABCDEFWXYZ**

**pattern = Right(teleno, 4)**

**If chkadjacent\_old(pattern) = 1 Then**

**chkvanity\_old = "ABCDEFWXYZ"**

**Exit Function**

**End If**

**'ABCDXYZXYZ**

**If Mid(teleno, 5, 3) = Mid(teleno, 8, 3) Then**

**chkvanity\_old = "ABCDXYZXYZ"**

**Exit Function**

**End If**

**'ABCDEEDCBA**

**If (Mid(teleno, 1, 1) = Mid(teleno, 10, 1)) And (Mid(teleno, 2, 1) = Mid(teleno, 9, 1)) And (Mid(teleno, 3, 1) = Mid(teleno, 8, 1)) And (Mid(teleno, 4, 1) = Mid(teleno, 7, 1)) And (Mid(teleno, 5, 1) = Mid(teleno, 6, 1)) Then**

**chkvanity\_old = "ABCDEEDCBA"**

**Exit Function**

**End If**

**'ABCDXABCDY**

**If Mid(teleno, 1, 4) = Mid(teleno, 6, 4) Then**

**chkvanity\_old = "ABCDXABCDY"**

**Exit Function**

**End If**

**'ABCDEFG000**

**If Mid(teleno, 8, 3) = "000" Then**

**chkvanity\_old = "ABCDEFG000"**

**Exit Function**

**End If**

**'ABCDEF1008**

**If Mid(teleno, 7, 4) = "1008" Then**

**chkvanity\_old = "ABCDEF1008"**

**Exit Function**

**End If**

**'ABCDEFG786**

**If Mid(teleno, 8, 3) = "786" Then**

**chkvanity\_old = "ABCDEFG786"**

**Exit Function**

**End If**

**'ABCDEFXXYY**

**If (chksame1(Mid(teleno, 7, 2)) = 1 And chksame1(Mid(teleno, 9, 2)) = 1) Then**

**chkvanity\_old = "ABCDEFXXYY"**

**Exit Function**

**End If**

**'ABCDEFXYXY**

**If Mid(teleno, 7, 2) = Mid(teleno, 9, 2) Then**

**chkvanity\_old = "ABCDEFXYXY"**

**Exit Function**

**End If**

**End Function**

**Function chksame1(tel) As Integer**

**chksame1 = 0**

**y = Mid(tel, 1, 1)**

**'MsgBox y**

**For i = 2 To Len(tel - 1)**

**If Mid(tel, i, 1) = y Then**

**chksame1 = 1**

**Else**

**chksame1 = 0**

**Exit Function**

**End If**

**Next**

**'MsgBox chksame**

**End Function**

**Function chksame\_even(tel) As Integer**

**chksame\_even = 0**

**If (tel Mod 2) = 0 Then**

**y = Mid(tel, 1, 1)**

**MsgBox y**

**For i = 2 To (Len(tel) - 1)**

**MsgBox "SS" & Mid(tel, i, 1)**

**'MsgBox y + 1**

**If CInt(Mid(tel, i, 1)) = y Then**

**chksame\_even = 1**

**Else**

**chksame\_even = 0**

**Exit Function**

**End If**

**'y = y + 1**

**Next**

**'MsgBox chksame**

**End If**

**End Function**

**Function chksame\_odd(tel) As Integer**

**chksame\_odd = 0**

**If (tel Mod 2) <> 0 Then**

**y = Mid(tel, 1, 1)**

**'MsgBox y**

**For i = 2 To (Len(tel) - 1)**

**'MsgBox Mid(tel, i, 1)**

**'MsgBox y + 1**

**If CInt(Mid(tel, i, 1)) = y Then**

**chksame\_odd = 1**

**Else**

**chksame\_odd = 0**

**Exit Function**

**End If**

**'y = y + 1**

**Next**

**'MsgBox chksame**

**End If**

**End Function**

**Function chkrptpttn2(tel) As Integer**

**chkrptpttn2 = 0**

**y = Mid(tel, 5, 2)**

**If (Mid(tel, 7, 2) = y) And (Mid(tel, 9, 2)) Then**

**chkrptpttn2 = 1**

**Else**

**chkrptpttn2 = 0**

**Exit Function**

**End If**

**End Function**

**Function chkrptpttn3(tel) As Integer**

**chkrptpttn3 = 0**

**y = Mid(tel, 2, 3)**

**If (Mid(tel, 5, 3) = y) And (Mid(tel, 8, 3) = y) Then**

**chkrptpttn3 = 1**

**Else**

**chkrptpttn3 = 0**

**Exit Function**

**End If**

**End Function**

**Function chkrptpttn4(tel) As Integer**

**chkrptpttn4 = 0**

**y = Mid(tel, 3, 4)**

**If Mid(tel, 7, 4) = y Then**

**chkrptpttn4 = 1**

**Else**

**chkrptpttn4 = 0**

**Exit Function**

**End If**

**End Function**

**Function chkrptpttn5(tel) As Integer**

**chkrptpttn5 = 0**

**If (Mid(tel, 1, 5) = Mid(tel, 6, 5)) Then**

**chkrptpttn5 = 1**

**Else**

**chkrptpttn5 = 0**

**Exit Function**

**End If**

**End Function**

**Function chkadjacent\_old(tel) As Integer**

**chkadjacent\_old = 0**

**y = Mid(tel, 1, 1)**

**'MsgBox y**

**For i = 2 To Len(tel - 1)**

**'MsgBox Mid(tel, i, 1)**

**'MsgBox y + 1**

**If CInt(Mid(tel, i, 1)) = y + 1 Then**

**chkadjacent\_old = 1**

**Else**

**chkadjacent\_old = 0**

**Exit Function**

**End If**

**y = y + 1**

**Next**

**'MsgBox chkadjacent**

**End Function**

**Function chkrptptnadj2(tel) As Integer**

**chkrptptnadj2 = 0**

**If (Mid(tel, 5, 1) = Mid(tel, 7, 1)) And (Mid(tel, 5, 1) = Mid(tel, 9, 1)) Then**

**If (CInt(Mid(tel, 6, 1) + 1) = CInt(Mid(tel, 8, 1))) And (CInt(Mid(tel, 8, 1) + 1) = CInt(Mid(tel, 10, 1))) Then**

**chkrptptnadj2 = 1**

**End If**

**Else**

**chkrptptnadj2 = 0**

**Exit Function**

**End If**

**End Function**