Republic of Iraq
University of Technology
Department of Computer Science

Block cipher lab

Computer and Cyber Security

2nd stage

Lecturer:
Fadhil Abbas Fadhil

Introduction:

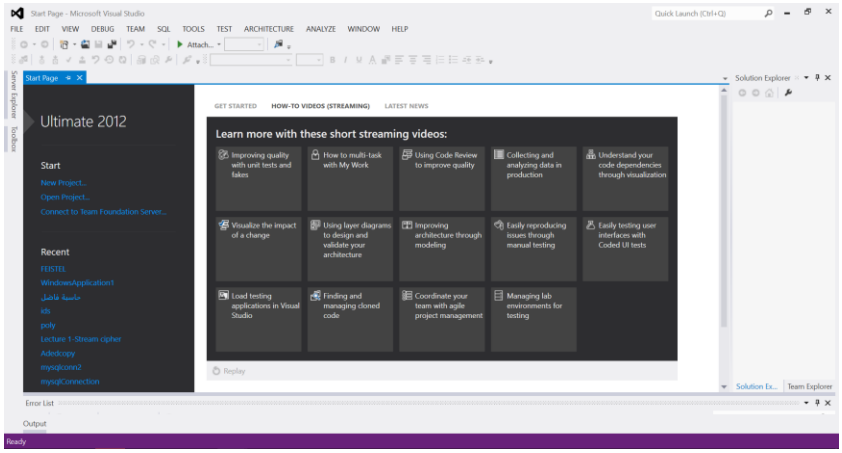
Visual Basic 2012 Tutorial?

Visual Basic 2012 was launched by Microsoft in 2012. Similar to the earlier versions of VB.NET programming languages, it is integrated with other Microsoft Programming languages in an IDE known as Visual Studio 2012.

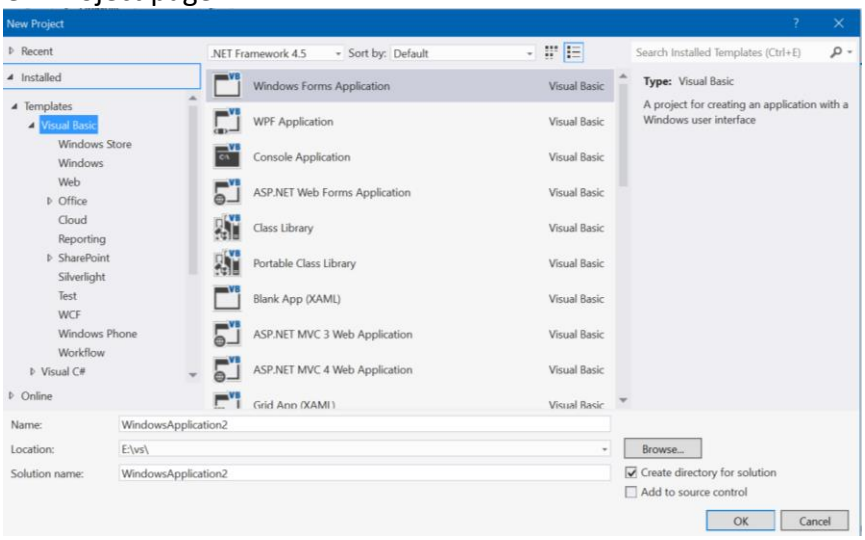
Although Microsoft had launched a few newer versions of Visual Studio until the latest Visual Studio 2017, you can still download the older version Visual Studio 2012 Express Edition from the following link:

https://www.visualstudio.com/vs/older-downloads/

When you launch Visual Studio Express 2012, the start page will appear, as shown in Figure below:

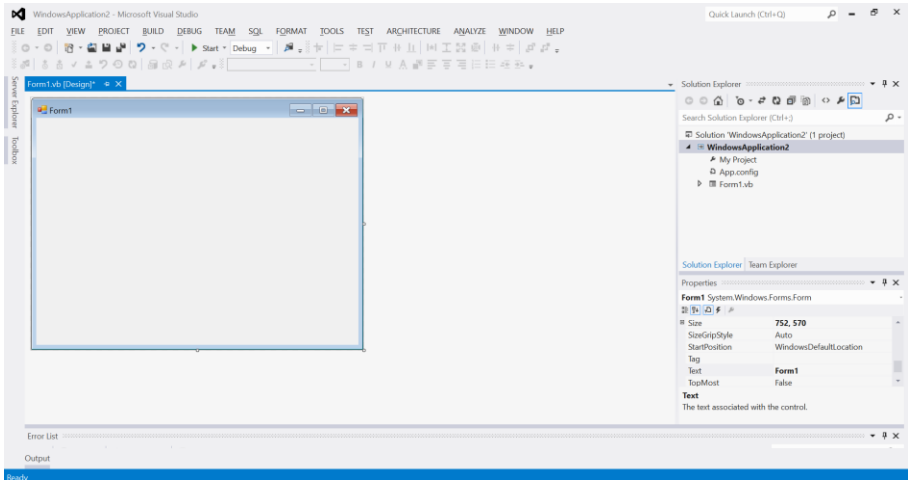


To start a new Visual Studio Express 2012 project, simply click on New Project to launch the Visual Studio New Project page

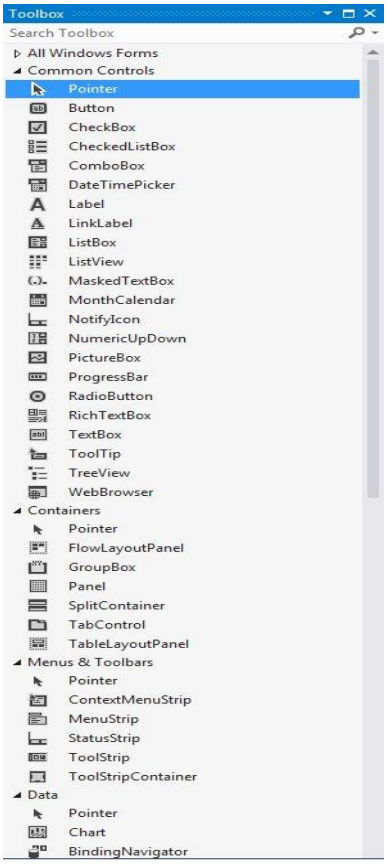


The New Project Page comprises three templates, Visual Basic, Visual C# and Visual C++. Since we are going to learn Visual Basic 2012, we shall select Visual Basic. Visual Basic 2012 offers you four types of projects that you can create. As we are going to learn to create Windows Applications, we will select Windows Forms Application.

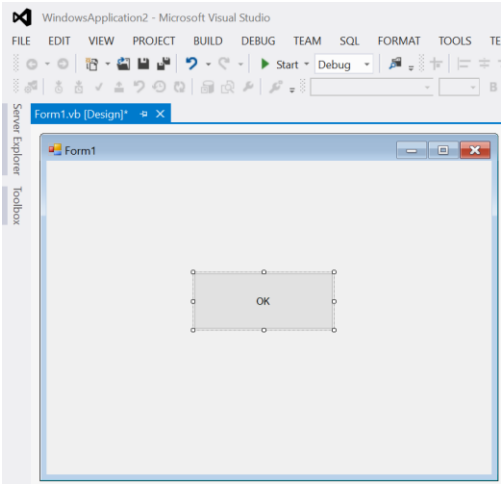
At the bottom of this dialog box, you can change the default project name WindowsApplication1 to some other name you like, for example, *WindowsApplications2*. After you have renamed the project, click OK to continue. The following IDE Windows will appear, it is similar to Visual Basic 2010. The Toolbox is not shown until you click on the Toolbox tab. When you click on the Toolbox tab, the common controls Toolbox will appear.



Visual Basic Express 2012 IDE comprises a few windows, the Form window, the Solution Explorer window and the Properties window. It also consists of a toolbox which contains many useful controls that allow a programmer to develop his or her VB programs.



Now, we shall proceed to show you how to create your first program. First, change the text of the form to My First Program in the properties window, it will appear as the title of the program. Next, insert a button and change its text to OK.

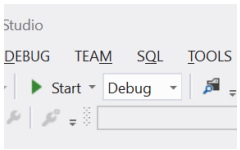


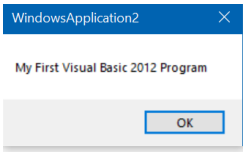
Now click on the OK button to bring up the code window and enter the following statement between Private Sub and End Sub procedure.

```
Public Class Form1
```

```
Private Sub Button1_Click(sender As Object, e As ventArgs) Handles Button1.Click
        MsgBox("My First Visual Basic 2012 Program")
    End Sub

End Class
```

Now click on the Start on the toolbar to run the program then click on the OK button, a dialog box that displays the “My First Visual Basic 2012 Program” message will appear.

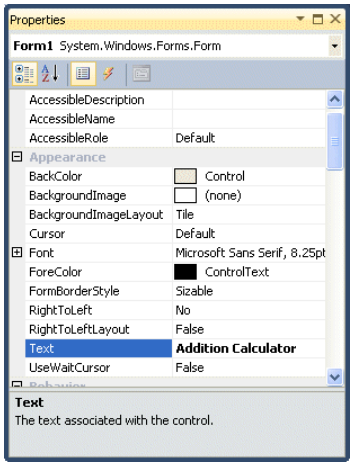




The function **MsgBox** is a *built-in function* of Visual Basic 2012 and it will display the text enclosed within the brackets.

The Control Properties

All controls in Visual Basic 2012 IDE have properties. By altering the properties of a control, we are able to customize its appearance and how it responds to an event. In the properties window, the item appears at the top part is the object currently selected. At the bottom part, the items listed in the left column represent the names of various properties associated with the selected object while the items listed in the right column represent the states of the properties. Properties can be set by highlighting the items in the right column then change them by typing or selecting the options available.



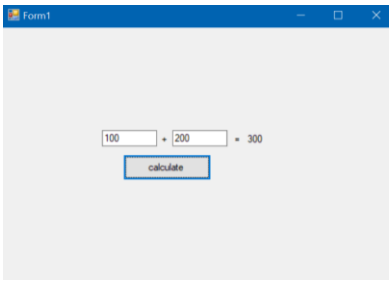
Inputs in Visual Basic 2012:

1. Using the **TextBox**

we will show you how to create a simple calculator that adds two numbers using the TextBox control. In this program, you insert two text boxes, three labels, and one button. The two text boxes are for the users to enter two numbers, one label is to display the addition operator and the other label is to display the equal sign. The last label is to display the answer. Now change the label on the button to Calculate, then click on this button and enter the following code:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim num1, num2, product As Single
    num1 = TextBox1.Text
    num2 = TextBox2.Text
    product = num1 + num2
    Label3.Text = product
End Sub
```

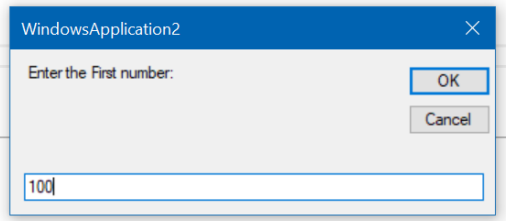
When you run the program and enter two numbers, pressing the calculate button adds the two numbers.

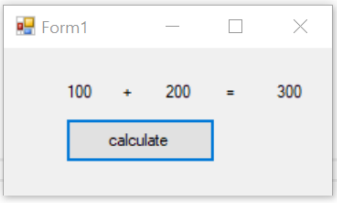


2. Using the **InputBox**

Using the same program we can use **InputBox** instead of TextBox:

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim num1, num2, product As Single
        num1 = InputBox("Enter the First number:")
        num2 = InputBox("Enter the Second number:")
        product = num1 + num2
        Label1.Text = num1
        Label2.Text = num2
        Label3.Text = product
    End Sub
End Class
```





Outputs in Visual Basic 2012:

1. Using the **Label**

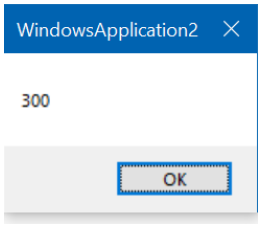
In the previous example, we notice the use of a label to display the results:

```
Label1.Text = num1
Label2.Text = num2
Label3.Text = product
```

2. Using the **Msgbox**

To display the results, we can also use the (**Msgbox**) function, as shown in the figure below:

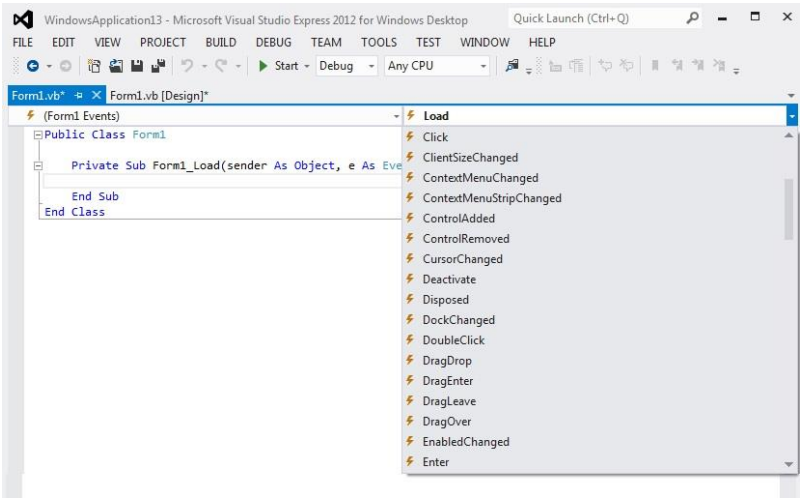
```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim num1, num2, product As Single
        num1 = InputBox("Enter the First number:")
        num2 = InputBox("Enter the Second number:")
        product = num1 + num2
        MsgBox(product)
    End Sub
End Class
```



line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

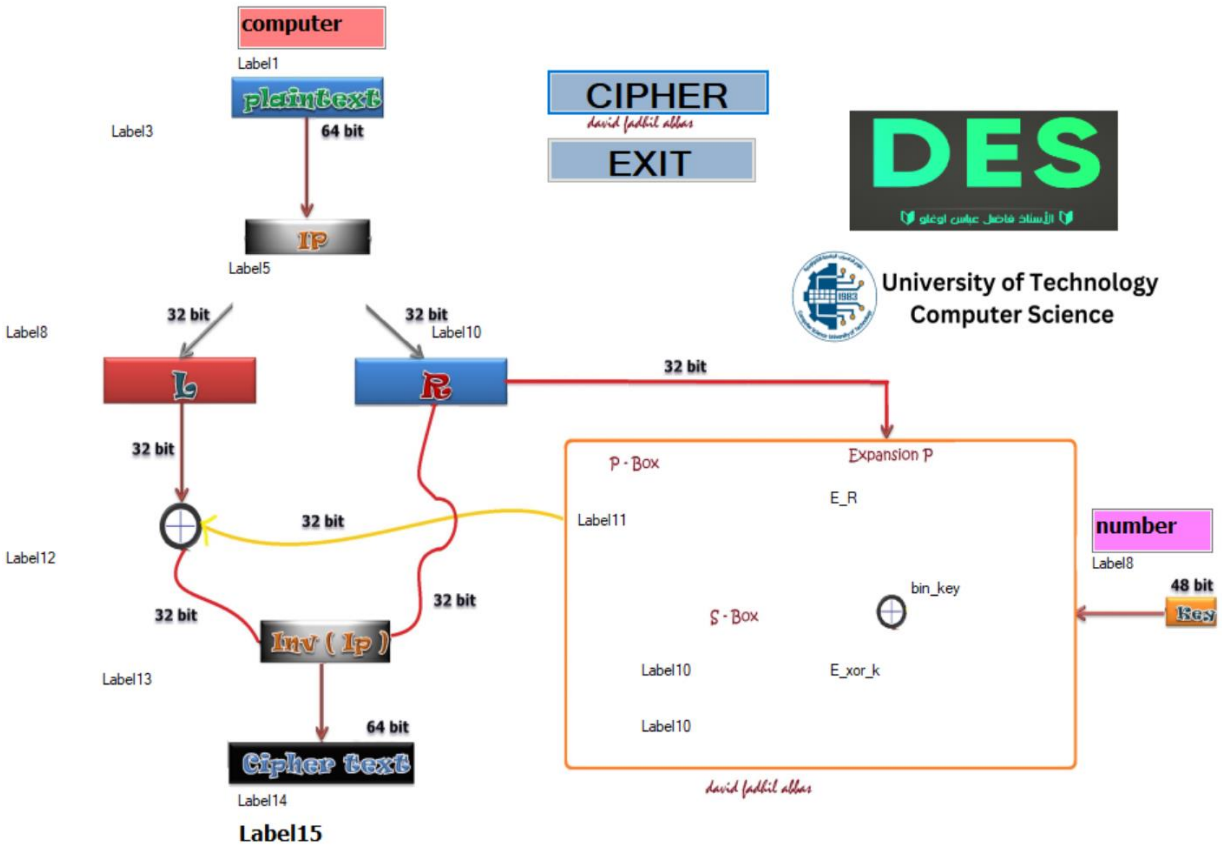
The Event Procedure

Each event is related to an object, it is an incident that happens to the object due to the action of the user. A class has events as it creates an instant of a class or an object. When we start a windows application in Visual Basic 2012, we will see a default form with the name Form1 appears in the IDE, it is actually the Form1 Class that inherits from the Form class **System.Windows.Forms.Form:**

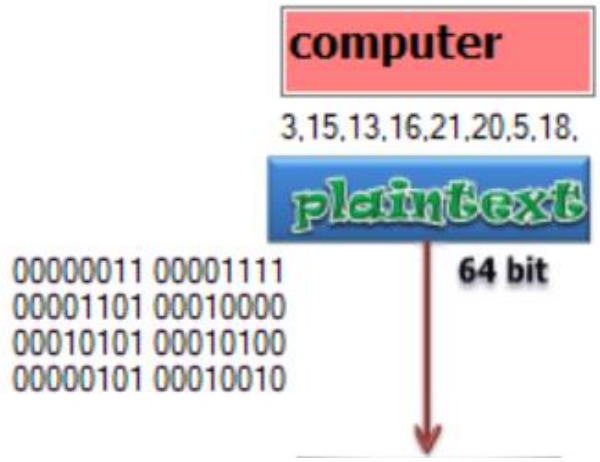


Lectures:

Initially, it's essential to grasp the fundamentals of encrypting data using the **DES** algorithm. To execute this cryptographic method effectively, it's imperative to develop multiple software functions necessary for encryption.

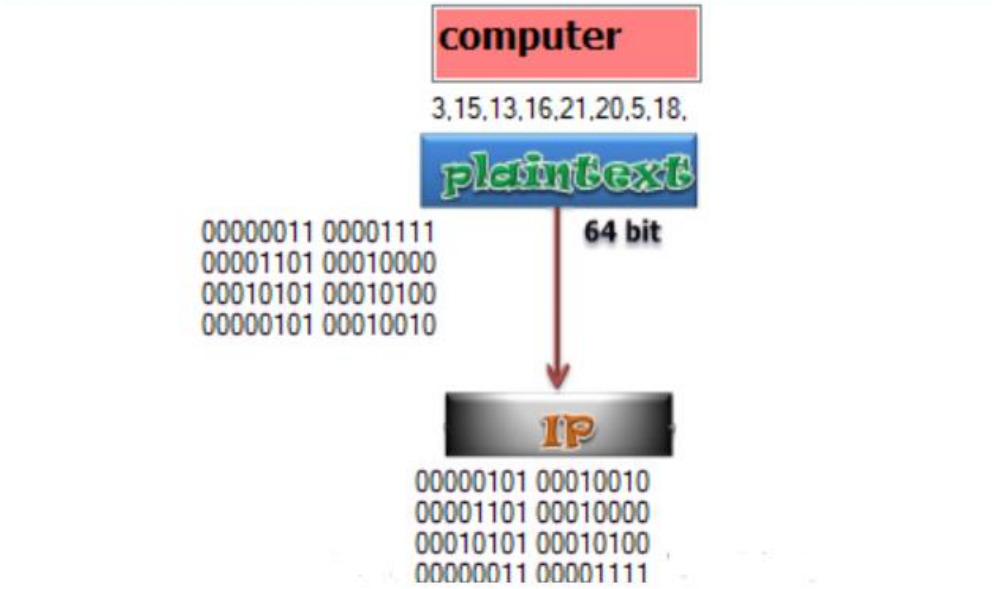


```
Private Function Convert_Plan_Tobin(ByVal str As String)
        Dim p As String = ""
        For i = 1 To Len(str)
            p = p & ASSCII(Mid(Plantext.Text, i, 1)) & ","
        Next

        plantext_ASCII.Text = p
        Dim bin_plan
        Dim a

        bin_plan = ""
        a = Split(p, ",")

        For i = LBound(a) To UBound(a) - 1
            Plantext_bin_label.Text = Plantext_bin_label.Text & Bin8bit(a(i)) & " "
            bin_plan = bin_plan & Bin8bit(a(i))
            If i Mod 2 <> 0 Then
                Plantext_bin_label.Text = Plantext_bin_label.Text & vbNewLine
            End If
        Next
        Return p
    End Function
```

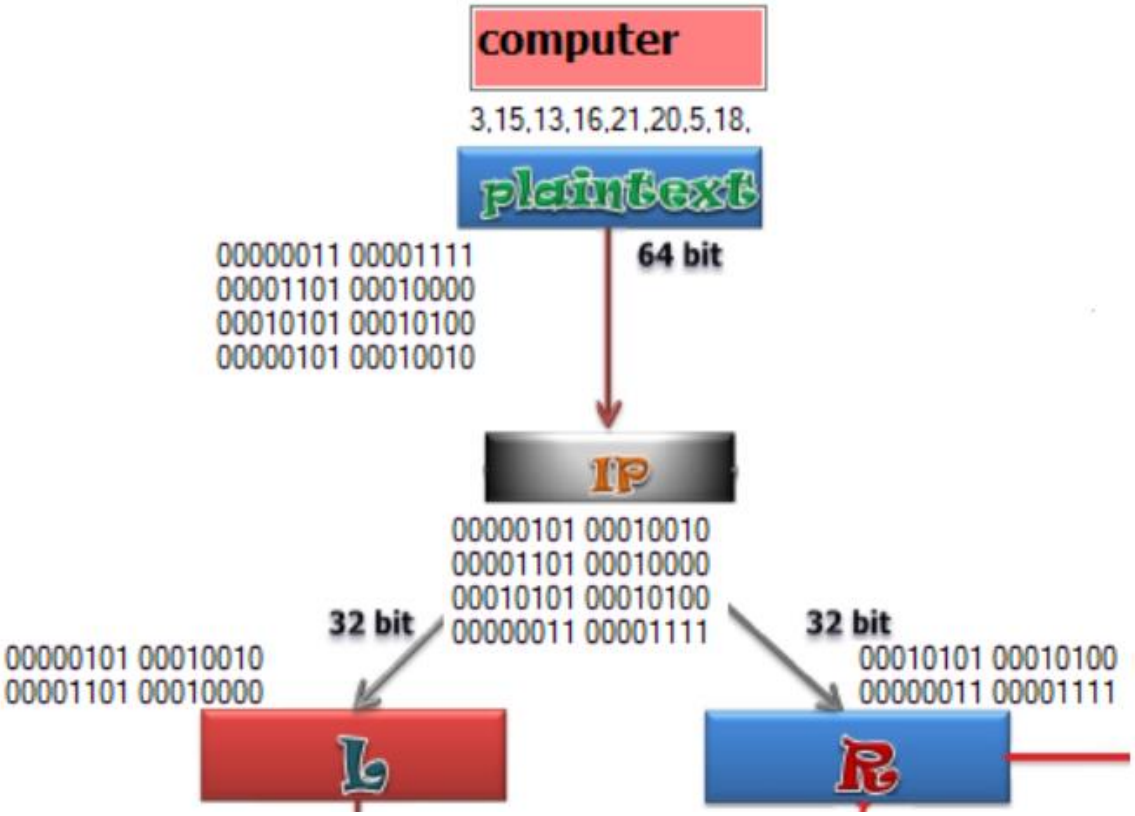


Then swap the positions using IP function:

```
    Private Function IP_finction(ByVal str As String)
        Dim a1, t
        a1 = Split(str, ",")
        t = a1(0) : a1(0) = a1(6) : a1(6) = t
        t = a1(1) : a1(1) = a1(7) : a1(7) = t
        Return a1
    End Function
```



Then split the result into Right and Left:

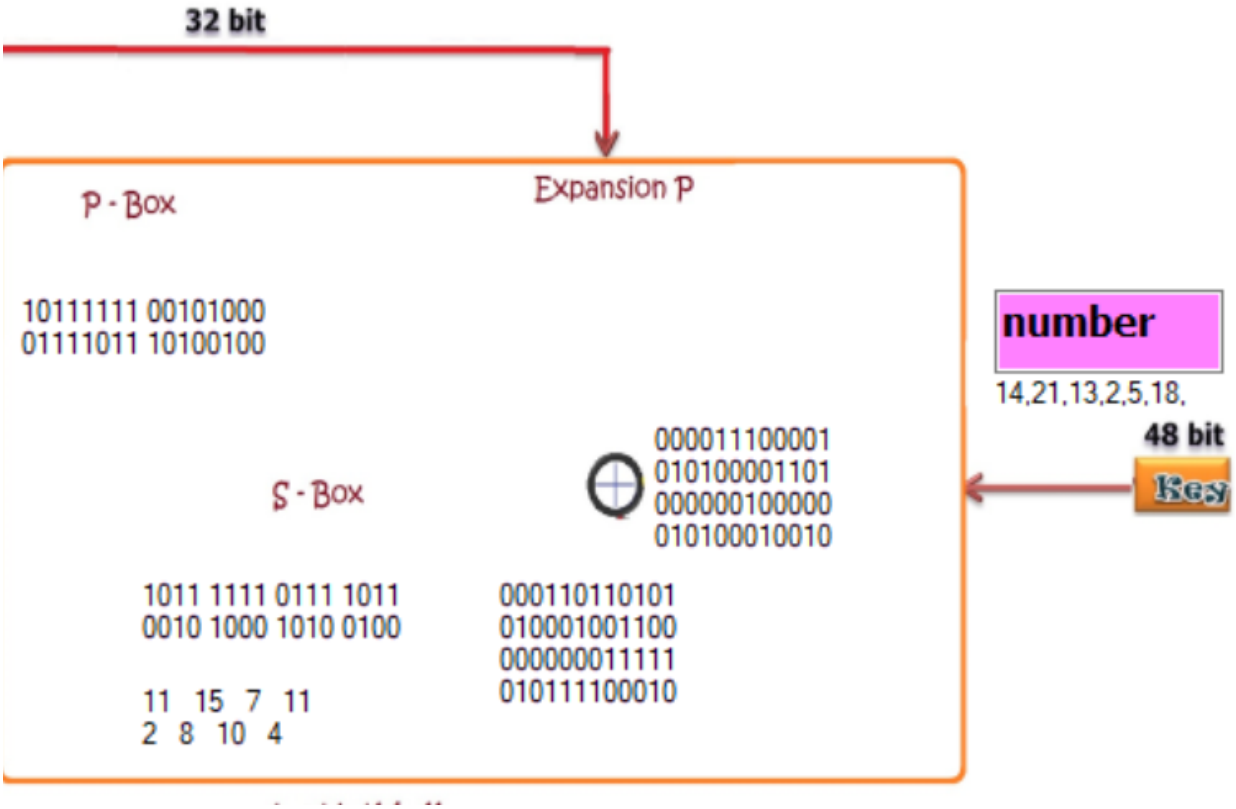



```
'------------------------------------------------------------  Find R and L
```

```
    For i = LBound(a_ip) To UBound(a_ip) - 1
        ip = ip & Bin8bit(a_ip(i))
        IP_label.Text = IP_label.Text & Bin8bit(a_ip(i)) & " "
        If i Mod 2 <> 0 Then
            IP_label.Text = IP_label.Text & vbNewLine
        End If
        If i <= 3 Then
            lef = lef & Bin8bit(a_ip(i))
            Left_label.Text = Left_label.Text & Bin8bit(a_ip(i)) & " "
            If i Mod 2 <> 0 Then
                Left_label.Text = Left_label.Text & vbNewLine
            End If
        Else
            rig = rig & Bin8bit(a_ip(i))
            Right_label.Text = Right_label.Text & Bin8bit(a_ip(i)) & " "
            If i Mod 2 <> 0 Then
                Right_label.Text = Right_label.Text & vbNewLine
            End If
        End If
    Next
```

After that we should compute F-function using key:

```
Private Function F_function(ByVal r As String, ByVal key As String)

    Dim ERR
    ERR = E_R_function(r)
    Dim e_xor_k
    e_xor_k = ""
    E_xor_k_label.Text = ""
    For i = 1 To Len(ERR)
        E_xor_k_label.Text = E_xor_k_label.Text & (Mid(ERR, i, 1) Xor Mid(key, i, 1))
        If i Mod 12 = 0 Then E_xor_k_label.Text = E_xor_k_label.Text & vbNewLine
        e_xor_k = e_xor_k & (Mid(ERR, i, 1) Xor Mid(key, i, 1))
    Next

    Dim s_box As String
    s_box = s_box_function(e_xor_k)
    Dim p_box As String
    p_box = p_box_function(s_box)
    Return p_box
End Function
```



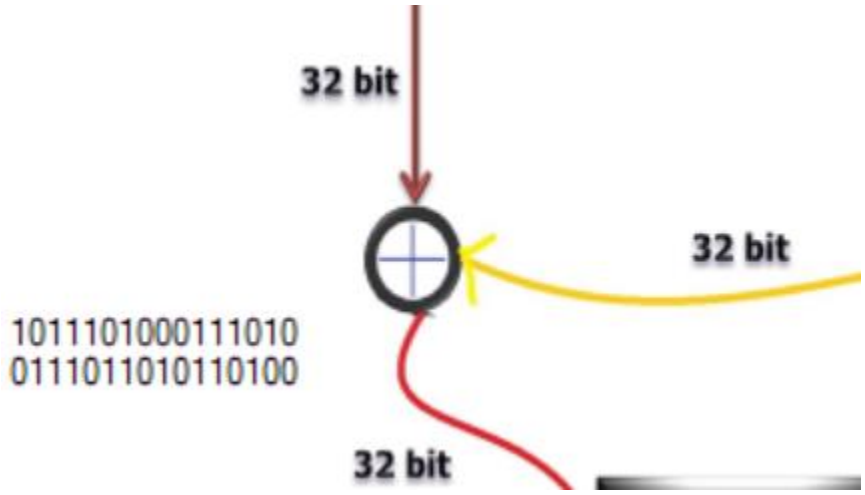

```
Private Function E_R_function(ByVal rig As String)
        Dim a3, ERR
        ERR = ""
        E_R.Text = ""
        a3 = Split(rig, " ")
        For i = LBound(a3) To UBound(a3) - 1

            If i Mod 2 = 0 Then
                E_R.Text = E_R.Text & a3(i) & Mid(a3(i + 1), 5, 4) & vbNewLine
            Else
                E_R.Text = E_R.Text & a3(i) & Mid(a3(i - 1), i, 4)
            End If
        Next
        ERR = Mid(rig, 1, 8) & Mid(rig, 13, 4) & Mid(rig, 9, 8) & Mid(rig, 1, 4) &
Mid(rig, 17, 8) & Mid(rig, 29, 4) & Mid(rig, 25, 8) & Mid(rig, 17, 4)
        Return ERR
    End Function

Private Function s_box_function(ByRef e_xor_k As String)
        Dim s_box = ""
        Dim k = 1, row, column
```

```
        Dim item = 0
        Dim s1(16, 16), s2(16, 16), s3(16, 16), s4(16, 16), s5(16, 16), s6(16, 16),
s7(16, 16), s8(16, 16)
        s1(0, 6) = 11 : s2(3, 5) = 15 : s3(1, 1) = 7 : s4(0, 12) = 11
        s5(0, 0) = 2 : s6(1, 15) = 8 : s7(1, 7) = 10 : s8(2, 2) = 4
        S_box_label.Text = ""
        S_box_label_int.Text = ""
        For i = 1 To Len(e_xor_k)
            row = integ(Mid(e_xor_k, i, 2))
            i += 2
            column = integ(Mid(e_xor_k, i, 4))
            i += 3
            Select Case k
                Case 1 : item = s1(row, column) : Case 2 : item = s2(row, column)
                Case 3 : item = s3(row, column) : Case 4 : item = s4(row, column)
                Case 5 : item = s5(row, column) : Case 6 : item = s6(row, column)
                Case 7 : item = s7(row, column) : Case 8 : item = s8(row, column)
            End Select
            s_box = s_box & Bin4bit(item)
            If k Mod 2 = 0 Then s_box = s_box & " "
            k += 1
            S_box_label_int.Text = S_box_label_int.Text & item & "    "
            S_box_label.Text = S_box_label.Text & Bin4bit(item) & " "
            If k = 5 Then
                S_box_label.Text = S_box_label.Text & vbNewLine
                S_box_label_int.Text = S_box_label_int.Text & vbNewLine

            End If

        Next

        Return s_box
    End Function

    Private Function p_box_function(ByRef s_box As String)
        Dim p_box
        Dim a4
        p_box = ""
        p_box_label.Text = ""
        a4 = Split(s_box, " ")
        p_box = p_box & a4(0) & a4(2) & a4(1) & a4(3)
        p_box_label.Text = p_box_label.Text & a4(0) & " " & a4(2) & vbNewLine & a4(1) &
" " & a4(3)

        Return p_box
    End Function
```

Then continue the algorithm steps:

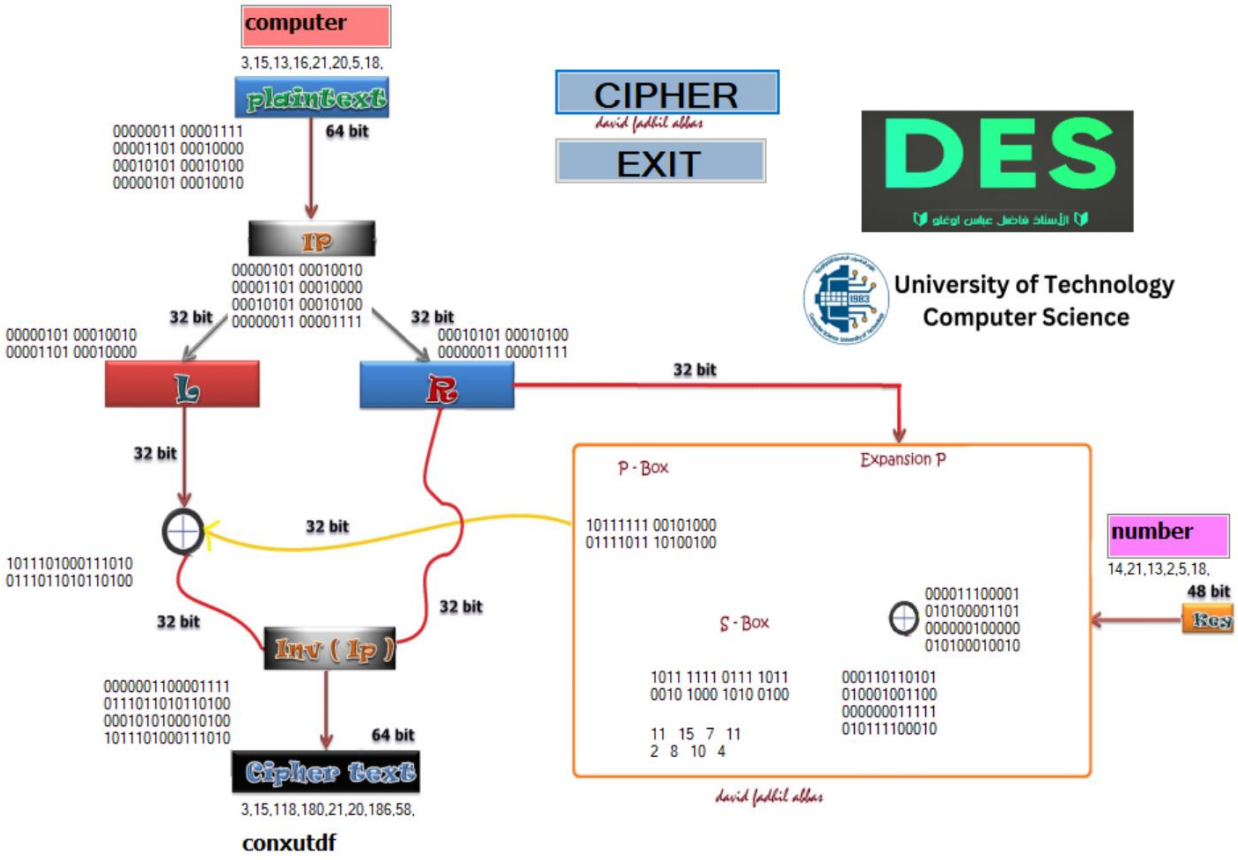
```
'------------------------------------------------------------ XOR
        Dim L_xor_Function
        L_xor_Function = ""
        Dim x = ""
        For i = 1 To Len(lef)
```

```
            x = Mid(lef, i, 1) Xor Mid(F, i, 1)
            L_xor_Function = L_xor_Function & x
            L_XOR_F_label.Text = L_XOR_F_label.Text & x
            If i Mod 16 = 0 Then L_XOR_F_label.Text = L_XOR_F_label.Text & vbNewLine
        Next
```




```
        '------------------------------------------------------------ inv_IP

        Dim inv_ip
        inv_ip = ""
        inv_ip = Mid(rig, 17, 16) & Mid(L_xor_Function, 17, 16) & Mid(rig, 1, 16) &
Mid(L_xor_Function, 1, 16)
        INV_IP_label.Text = Mid(rig, 17, 16) & vbNewLine & Mid(L_xor_Function, 17, 16) &
vbNewLine & Mid(rig, 1, 16) & vbNewLine & Mid(L_xor_Function, 1, 16)
```


```
        '------------------------------------------------------------  ASCII Convertion
        Dim ASCII_cipher
        ASCII_cipher = ""
        For i = 1 To Len(inv_ip) Step 8
            ASCII_cipher = ASCII_cipher & integ(Mid(inv_ip, i, 8)) & ","
        Next
        REV_Bin_label.Text = ASCII_cipher
        Dim a5
        a5 = Split(ASCII_cipher, ",")
        For i = LBound(a5) To UBound(a5) - 1
            ciphertext_label.Text = ciphertext_label.Text & tochar_ASSCII(a5(i)) Mod 26)
        Next
```


computer
3,15,13,16,21,20,5,18,
plaintext
64 bit
00000011 00001111
00001101 00010000
00010101 00010100
00000101 00010010
IP
00000101 00010010
00001101 00010000
00010101 00010100
00000011 00001111
32 bit
32 bit
00000101 00010010
00001101 00010000
00010101 00010100
00000011 00001111
L
R
32 bit
32 bit
32 bit
1011101000111010
0111011010110100
32 bit
32 bit
Inv (Ip)
0000001100001111
0111011010110100
0001010100010100
1011101000111010
64 bit
Cipher text
3,15,118,180,21,20,186,58,
conxutdf
CIPHER
david fadhil abbas
EXIT
DES
University of Technology
Computer Science
P - Box
Expansion P
10111111 00101000
01111011 10100100
S - Box
1011 1111 0111 1011
0010 1000 1010 0100
11 15 7 11
2 8 10 4
000011100001
010100001101
000000100000
010100010010
000110110101
010001001100
000000011111
010111100010
number
14,21,13,2,5,18,
48 bit
Key
david fadhil abbas