Objectives

Program Purpose

• Analyse numbers for primeness and factors
• List all primes up to a given number
• List all factors of a given number

Learning Goals

• Use of Functions and User-Defined Functions
• Use of Modulus
• More complex algorithms
• Use of ReDim to dynamically declare array size

Design Notes

This program is more of a challenge than the previous ones. To determine if a number is prime, we need to check each integer up to the square root of the number to see if it is a factor. If the number 1 is the only factor, then the original number is prime.

There are many websites devoted to the challenge of finding bigger and bigger prime numbers. It is an interesting mathematical challenge, and as computing power increases, the boundaries are continually pushed further back. Maybe you will discover the largest prime known to humanity!

A function is different from a normal procedure in that it returns a value. In this project we create our own functions:

a. Square() - which determines if a number is a Square number
b. Prime() - which determines if a number is Prime.

We also use the pre-defined Function Sqr(), which determines the Square root of a number.

The Mod operator is used throughout to test for remainders and hence to find out if a number is a factor or not. e.g. 10 Mod 3 = 1 - the remainder after dividing 10 by 3 is 1, therefore 3 is not a factor of 10.

There are many ways to improve the coding used in this project. For teaching purposes, we have tried to keep things relatively simple. You may find more efficient methods. Good luck!

Interface

Create the interface as shown.
Use 1 form, 2 list boxes, 7 labels, 1 text box and 1 command button.
Names of Objects

<table>
<thead>
<tr>
<th>Type of Object</th>
<th>Number</th>
<th>Names of Objects</th>
<th>Simple Initial Properties of Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>1</td>
<td>Form1</td>
<td>Caption - &quot;Prime Number Checker 1&quot;</td>
</tr>
<tr>
<td>List Box</td>
<td>2</td>
<td>lstFactors, lstPrimes</td>
<td>Font - Bold, 12 Columns - 5</td>
</tr>
<tr>
<td>Command Buttons</td>
<td>1</td>
<td>cmdCheck</td>
<td>Font Bold, 12 Caption - &amp;Check Prime</td>
</tr>
<tr>
<td>Test Box</td>
<td>1</td>
<td>Text1</td>
<td>Font - Bold, 12 Text - 7560</td>
</tr>
<tr>
<td>Labels</td>
<td>7</td>
<td>lblPrime, lblComposite</td>
<td>Font - Bold, 12 Visible - False Caption &quot;Prime&quot; BackColor - Green Backstyle - Opaque</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lblTimeTaken, lblLastPrime, lblNumFactors</td>
<td>Font - Bold, 12 Captions &quot;&quot; Borderstyle - Fixed Single Backstyle - Opaque</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Label1, Label2</td>
<td>Caption - &quot;Highest Prime&quot; Caption - &quot;Primes Up to That Number&quot;</td>
</tr>
</tbody>
</table>

Code

**GENERAL SECTION**

Dim Primes() As Long
Dim NumPrimes As Long
Const ProgramLimit = 385639

Sub wait(x As Single)
Dim y As Single
y = Timer
While x + y > Timer
    DoEvents
Wend
End Sub

Sub addtoprimearray(primeNum As Long)
    NumPrimes = NumPrimes + 1
    ReDim Preserve Primes(NumPrimes)
    Primes(NumPrimes) = primeNum
End Sub

Sub CheckPrimes(Limit As Long)
Dim n As Long
n = 1
While n <= Limit And n < ProgramLimit
    If Prime(n) Then
        lblLastPrime.Caption = n
        lstPrimes.AddItem n
        addtoprimearray n
    End If
    n = n + 1
Wend
If Prime(Limit) Then
    lblPrime.Visible = True
    lblComposite.Visible = False
Else
    lblPrime.Visible = False
    lblComposite.Visible = True
End If
End Sub
Function Prime(n As Long) As Integer

    Dim i As Integer
    Prime = True
    i = 2
    While Prime And i <= Sqr(n)
        If n = Primes(i) Then
            Prime = True
        Else
            If n Mod i <> 0 Then
                If Not Square(n) Then
                    i = i + 1
                    Prime = False
                Else
                    Prime = False
                End If
            Else
                Prime = False
            End If
        End If
    Wend
End Function

Function Square(x As Long) As Boolean

    If Sqr(x) = Int(Sqr(x)) Then
        Square = True
    Else
        Square = False
    End If
End Function

Private Sub DisplayFactors(max As Integer)

    Dim i As Integer
    lblNumFactors.Caption = 0
    For i = 1 To max
        If max Mod i = 0 Then
            lstFactors.AddItem i
            lblNumFactors.Caption = lblNumFactors.Caption + 1
        End If
    Next i
End Sub
**EVENTS**

Private Sub cmdCheck_Click()
Dim start As Single
MousePointer = 11
start = Timer
lstPrimes.Clear
lstFactors.Clear
CheckPrimes Val(Text1.Text)
lblTimeTaken.Caption = Format(Timer - start, "###.###") & " sec"
DisplayFactors (Val(Text1.Text))
MousePointer = 0
End Sub

Private Sub Form_Load()
NumPrimes = 0
End Sub

Private Sub Text1_Change()
If Trim(Text1.Text) <> "" Then
If Val(Text1.Text) < 1 Or Val(Text1.Text) > ProgramLimit Then
    Beep
    lblTimeTaken.Caption = "Outside Limits"
    Text1.Locked = True
    wait 2
    lblTimeTaken.Caption = ""
    Text1.Text = ""
    Text1.Locked = False
End If
End If
End Sub

**Consolidation & Extension**

1. Which procedures are Functions?
2. Which built-in Functions are used?
3. What does the instruction 'MousePointer = 11' do?
4. What is Mod used for?
5. What does the 'text1_Change' Event do?
6. Explain in plain English the function of the 'CheckPrime' Algorithm.
7. Explain the function of the 'Square' Algorithm.
8. Some of the code is not very efficient. Which sections could be optimised?