**NAME: KYRSHANLANG R DKHAR**

**CLASS: MCA 3rd SEMESTER**

**SUBJECT: 305 LAB RECORD VB.NE**

Contents

[Console Applications 4](#_Toc463117514)

[1. Program to Find Factorial of a number 4](#_Toc463117515)

[2. Program to Find Sum of Fibonacci series upto n 5](#_Toc463117516)

[3. program to find the reverse of a number 6](#_Toc463117517)

[4. program to find the sum of the digits of a number 7](#_Toc463117518)

[5. program to check whether the number is a prime number 8](#_Toc463117519)

[6. program to check whether the number it is an armstrong number 9](#_Toc463117520)

[7. program to find all the factors of a number 10](#_Toc463117521)

[8. program to find a series of Armstrong numbers from 1 to ‘n ’ 11](#_Toc463117522)

[9. Program to find a series of prime numbers from 2 to ‘n’ 12](#_Toc463117523)

[10. Program to find whether a number ‘n’ is an Automorphic number or not 13](#_Toc463117524)

[ARRAY CONSOLE APPLICATIONS 14](#_Toc463117525)

[1. Program to Display sum and average of all elements in an array. 14](#_Toc463117526)

[2. Program to Display the values in the array, along with the largest and smallest element in the array. 15](#_Toc463117527)

[3. Program to perform searching for an element in an array. 15](#_Toc463117528)

[4. Program to store numeric values in an ‘m x n’ matrix and display the matrix and the sum of its diagonal elements. 15](#_Toc463117529)

[5. Program to store numeric values in two ‘m x n’ matrix and display the two matrices along with their sum and difference 15](#_Toc463117530)

[ARRAY GUI APPLICATIONS 16](#_Toc463117531)

[1. Program to Display the values in the array, the sum and average of all the elements in the array. 16](#_Toc463117532)

[2. Program to Display the values in the array, along with the largest and smallest element in the array 16](#_Toc463117533)

[3. Program to Display the values in the array, as well as, the result of the search 16](#_Toc463117534)

[4. Program to store numeric values in an ‘m x n’ matrix and display the matrix and the sum of its diagonal elements 16](#_Toc463117535)

[Dynamic Forms Worksheet 16](#_Toc463117536)

[1. Application on Quadratic Equation and display its roots 16](#_Toc463117537)

[Class Worksheet 17](#_Toc463117538)

[Windows Application that uses methods of the class Triangle to find the area of a triangle. 17](#_Toc463117539)

[Windows Application that uses methods of the class to find the number of digits, sum of the digits of a number and reverse of the number. 17](#_Toc463117540)

[Class Worksheet on Strings 18](#_Toc463117541)

[Windows Application that uses methods of a class to determine if an inputted word is a palindrome. 18](#_Toc463117542)

[Windows Application that uses methods of a class to display the number of words, number of vowels, number of digits and number of special symbols present in a sentence. 19](#_Toc463117543)

[Windows Application that uses methods of a class to display whether the words entered by the user are Anagrams of each other. 20](#_Toc463117544)

[Numeric Class Worksheet 21](#_Toc463117545)

[Windows Application that uses methods of the class Armstrong to determine if a number is an Armstrong number or not. 21](#_Toc463117546)

[Windows Application that uses methods of the class Factors to find the factors of a number 21](#_Toc463117547)

# 

# Console Applications

## Program to Find Factorial of a number

Q1. Write a console application program to find the factorial of a number ‘n’, where ‘n’ is taken as input at run-time. Do an error-checking (using if-else statement) to check that the value of ‘n’ cannot be more than 20.

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n As Integer, i As Short, fact As Long

Console.WriteLine("Factorial Program")

Console.Write("Enter A Number: ")

n = Console.ReadLine()

fact = 1

If n >= 20 Then

Console.WriteLine("Number too Large")

Else

For i = 1 To n

fact = fact \* i

Next

Console.WriteLine("Factorial of : " & n & " is " & fact)

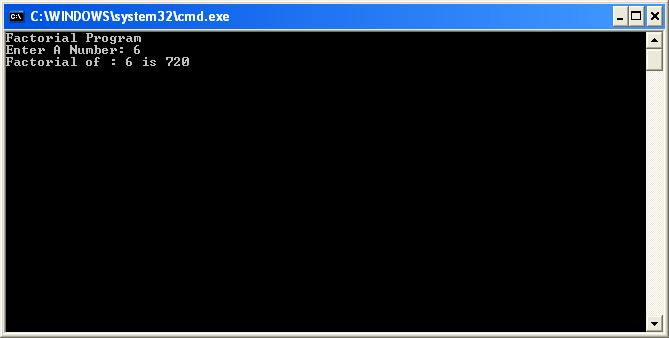
End If

Console.Read()

End Sub

End Module

**OUTPUT**



## Program to Find Sum of Fibonacci series upto n

Q2. Write a console application program to find the sum of the Fibonacci series up to ‘n’ number of terms, where ‘n’ is taken as input at run-time. For example, if the n=5, the output should be 7. (i. e. 0+1+1+2+3)

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n As Integer, sum As Decimal, f As Decimal, s As Decimal, nxt As Decimal

Console.WriteLine("Program to Find the sum of Fibonacci series")

Console.Write("Enter the limit: ")

n = Console.ReadLine()

f = 0

s = 1

sum = 0

nxt = 0

Dim i As Integer = 1

Console.WriteLine("Series is ")

While i <= n

sum += f

nxt = f + s

f = s

s = nxt

i += 1

'Console.WriteLine(" i " & i & " sum nxt f s " & sum & " " & nxt & " " & f & " " & s)

'Console.Read()

Console.Write(" " & f)

End While

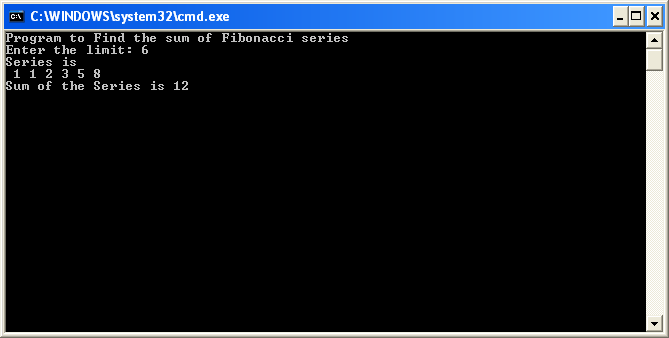
Console.WriteLine(vbNewLine & "Sum of the Series is " & sum)

Console.Read()

End Sub

End Module

**OUTPUT**



## program to find the reverse of a number

Write a console application program to find the reverse of a number ‘n’, where ‘n’ is taken as input at run-time. Also, provide a message whether the reversed number is equal to the original number or not.

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n As Integer, rev As Integer, r As Integer

Dim tn As Integer

Console.WriteLine("Program to Reverse a number")

Console.Write("Enter a Number : ")

n = Console.ReadLine()

rev = 0

tn = n

While tn > 0

r = tn Mod 10

rev = rev \* 10 + r

tn /= 10

End While

Console.WriteLine("Reverse no: " & rev)

If rev = n Then

Console.WriteLine(" The Reverse is equal to the Original Number")

Else

Console.WriteLine("The Reverse is not equal to the Original number")

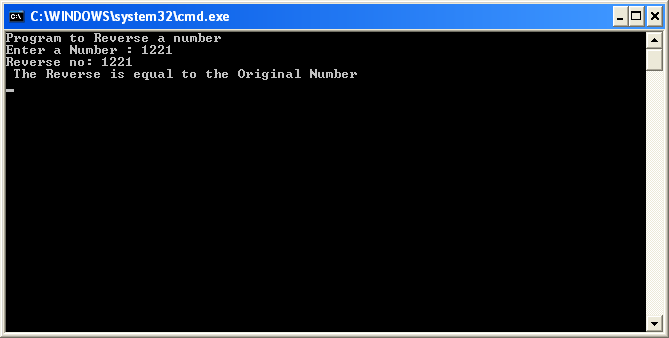
End If

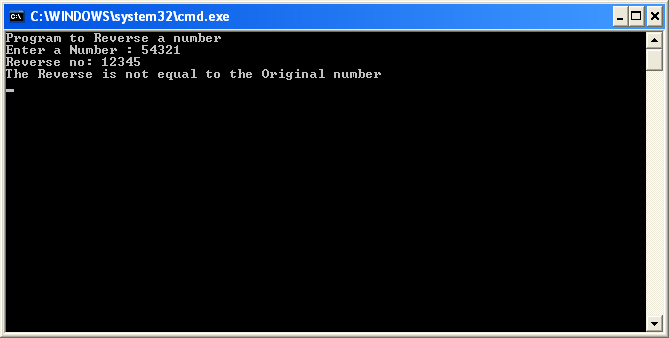
Console.Read()

End Sub

End Module

**OUTPUT**





## program to find the sum of the digits of a number

Write a console application program to find the sum of the digits of a number ‘n’, where ‘n’ is taken as input at run-time.

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n As Integer, rev As Integer, r As Integer

Dim tn As Integer

Console.WriteLine("Program to Find the sum of the digits of a number")

Console.Write("Enter a Number : ")

n = Console.ReadLine()

rev = 0

tn = n

While tn > 0

r = tn Mod 10

rev = rev + r

tn /= 10

End While

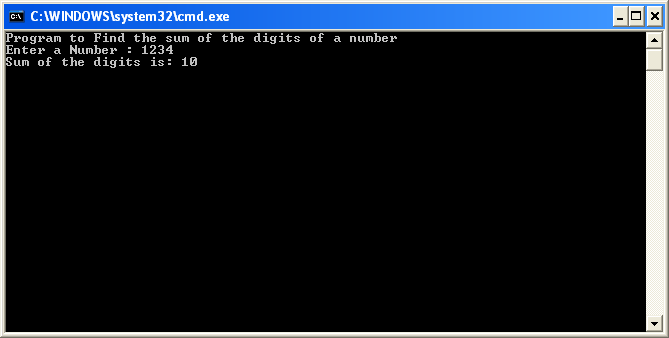
Console.WriteLine("Sum of the digits is: " & rev)

Console.Read()

End Sub

End Module

**OUTPUT**



## program to check whether the number is a prime number

Write a console application program to enter a number ‘n’ and check whether the number is a prime number or not. Display the message accordingly

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n As Integer, i As Integer, Flag As Boolean

Console.WriteLine("Program to Find Whether the number is Prime or Not")

Console.Write("Enter any Number : ")

n = Console.ReadLine

Flag = True

If n <> 2 Then

For i = 2 To n / 2

If n Mod i = 0 Then

Flag = False

End If

If Flag = False Then

Exit For

End If

'Console.WriteLine("i " & i)

Next

End If

If Flag = False Then

Console.WriteLine("It is not a Prime Number")

Else

Console.WriteLine("It is a Prime Number")

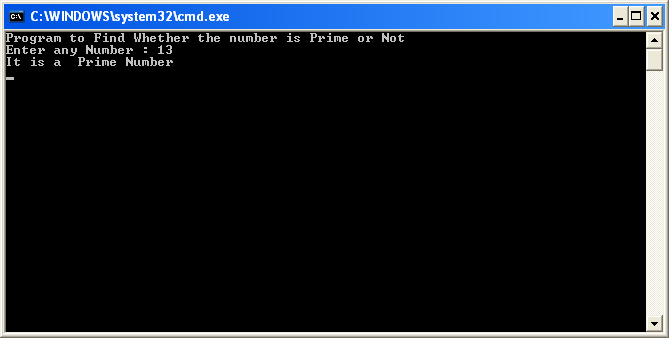
End If

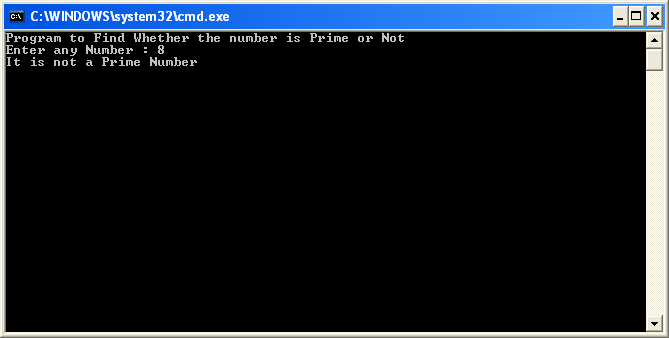
Console.ReadLine()

End Sub

End Module

**OUTPUT**





## program to check whether the number it is an armstrong number

Write a console application program to find enter a number ‘n’ and check whether the number it is an armstrong number or not . Display the message accordingly.

For example, if n=153, output is *“Armstrong Number”*

153 = (1\*1\*1)+(5\*5\*5)+(3\*3\*3)

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n As Integer, rev As Integer, r As Integer, tn As Integer

Console.WriteLine("Program to Check if a Number is an Armstrong Number")

Console.Write("Enter a Number : ")

n = Console.ReadLine()

rev = 0

tn = n

While n > 0

r = n Mod 10

rev = rev + r ^ 3

n = n \ 10

'Console.WriteLine(" r n " & n)

End While

' Console.WriteLine("Reverse no: " & rev)

If rev = tn Then

Console.WriteLine(" The Number is an Armstrong Number ")

Else

Console.WriteLine("The Number is not an Armstrong Number")

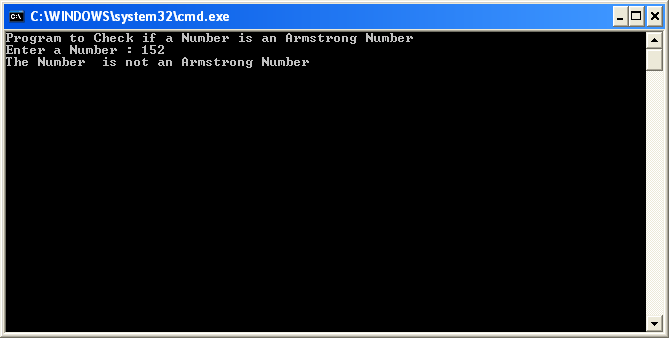
End If

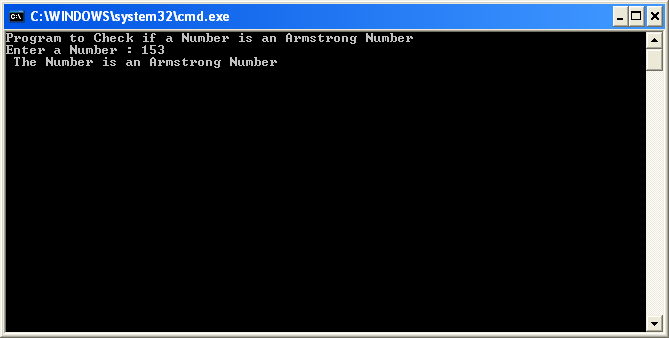
Console.Read()

End Sub

End Module

**OUTPUT**





## program to find all the factors of a number

1. Write a console application program to find all the factors of a number ‘n’, where ‘n’ is taken as input at run-time. For example, if n=10, output is 1, 2, 5, 10.

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim i, fac, n As Integer

Console.WriteLine("Program to Find FACTORS OF A NUMBER")

Console.Write("Enter a number: ")

n = Console.ReadLine()

Console.WriteLine("Factors of " & n & ":")

For i = 1 To n

If n Mod i = 0 Then

Console.WriteLine(" " & i)

End If

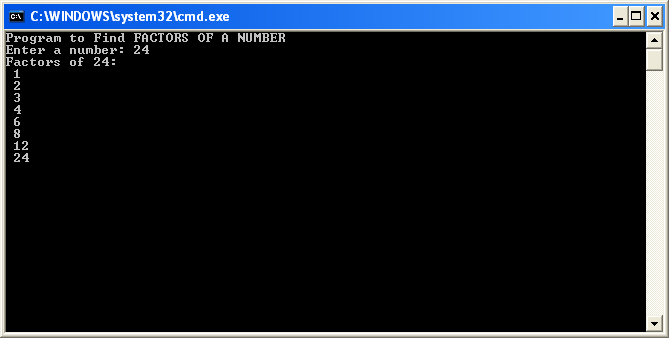
Next

Console.ReadKey()

End Sub

End Module

**OUTPUT**



## program to find a series of Armstrong numbers from 1 to ‘n ’

1. Write a console application program to find a series of Armstrong numbers from 1 to ‘n ’, where ‘n’ is taken as input at run-time.

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n As Integer, res As Integer, r As Integer

Dim tn As Integer

Console.WriteLine("Program to Generate Armstrong Numbers")

Console.Write("Enter a Limit: ")

n = Console.ReadLine()

For i = 1 To n

res = 0

tn = i

While tn > 0

r = tn Mod 10

res = res + r ^ 3

tn = tn \ 10

'Console.WriteLine(" r n " & n)

End While

' Console.WriteLine("Reverse no: " & rev)

If res = i Then

Console.WriteLine(" " & i)

End If

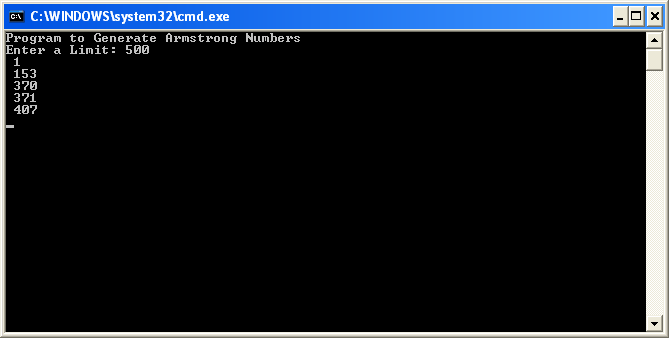
Next i

Console.Read()

End Sub

End Module

**OUTPUT**



## Program to find a series of prime numbers from 2 to ‘n’

1. Write a console application program to find a series of prime numbers from 2 to ‘n’, where ‘n’ is taken as input at run-time.

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n As Integer, i, tn, j As Integer, Flag As Boolean

Console.WriteLine("Program to Generate Prime Number")

Console.Write("Enter Limit : ")

n = Console.ReadLine

Flag = True

For j = 2 To n

tn = j

'If tn <> 2 Then

For i = 2 To j \ 2

If tn Mod i = 0 Then

Flag = False

Exit For

End If

'Console.WriteLine("i " & i)

Next i

' End If

'Console.WriteLine("" & j)

If Flag = False Then

Console.Write("")

Flag = True

Else

Console.WriteLine(" " & tn)

End If

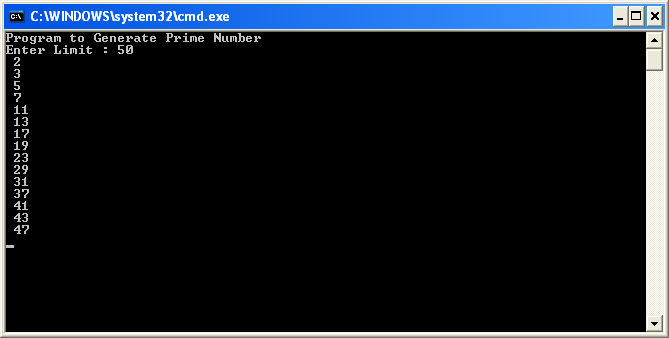
Next j

Console.ReadKey()

End Sub

End Module

**OUTPUT**



## Program to find whether a number ‘n’ is an Automorphic number or not

1. Write a console application program to find whether a number ‘n’ is an Automorphic number or not. Check that the number should only be a single digit number.

An automorphic number is a number which has if the last digit of the square of the number is same as the number itself.

Example: **6** is automorphic since 6\*6=3**6**

**5** is automorphic since 5\*5=2**5**

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n, Sq, last As Integer

Console.WriteLine("Program to Check For an Automorphic Number")

Console.WriteLine("Enter a Number : ")

n = Console.ReadLine()

If n > 9 Or n < 1 Then

Console.WriteLine("Invalid input")

Else

Sq = n \* n

last = Sq Mod 10

If last = n Then

Console.WriteLine("It is Automorphic")

Else

Console.WriteLine("It is Not an automorphic number")

End If

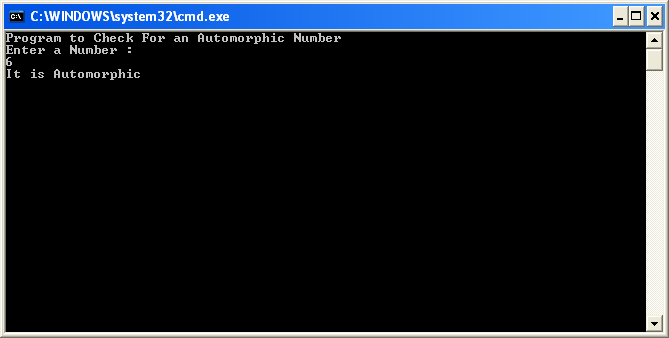
End If

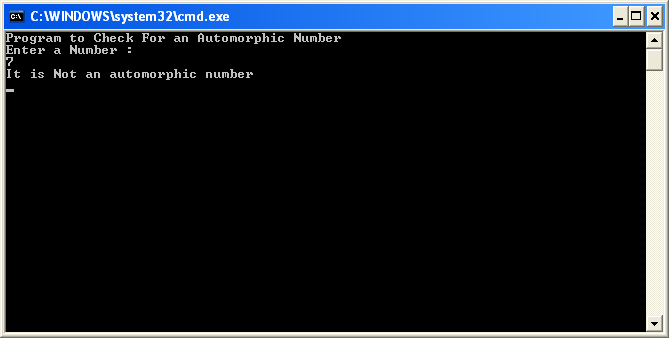
Console.Read()

End Sub

End Module

**OUTPUT**





# ARRAY CONSOLE APPLICATIONS

## Program to Display sum and average of all elements in an array.

1. Write a Console Application in VB that will as input the number of numeric values in an array. Display the values in the array, as well as, the sum and average of all the elements in the array.

Soln: Module1.vb ‘ 01Array

Module Module1

Sub Main()

Dim n, Array(100) As Integer

Dim sum As Integer, avg As Decimal

Console.WriteLine("Enter the size to the array")

n = Console.ReadLine()

Console.WriteLine("Enter the values to the array")

For i = 0 To n - 1

Array(i) = Console.ReadLine()

sum = sum + Array(i)

Next i

Console.WriteLine("The Array is :")

For i = 0 To n - 1

Console.WriteLine(" A(" & i & ") : " & Array(i))

Next

avg = sum / n

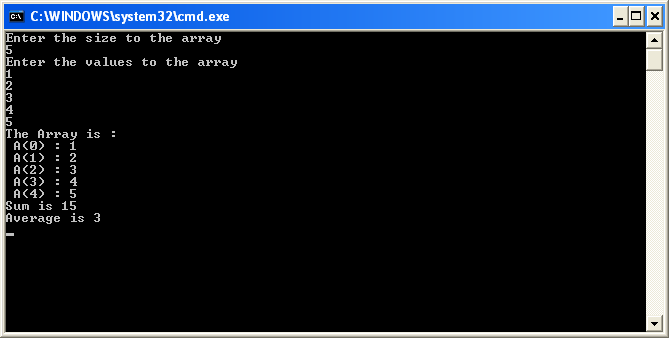
Console.WriteLine("Sum is " & sum)

Console.WriteLine("Average is " & avg)

Console.Read()

End Sub

End Module

**OUTPUT**

## Program to Display the values in the array, along with the largest and smallest element in the array.

2. Write a Console Application in VB that will as input the number of numeric values in an array. Display the values in the array, along with the largest and smallest element in the array. (Use subroutine for inputting values and displaying the array elements, function to return the largest and function to return the smallest without sorting the array elements)

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n, Array(100) As Integer

Dim lar, sml As Integer

Console.WriteLine("Enter the size to the array")

n = Console.ReadLine()

Call Input(Array, n)

lar = largest(Array, n)

sml = smallest(Array, n)

Call Display(Array, n, lar, sml)

Console.Read()

End Sub

Sub Input(ByRef A() As Integer, ByVal n As Integer)

Console.WriteLine("Enter the values to the array")

For i = 0 To n - 1

A(i) = Console.ReadLine()

Next i

End Sub

Sub Display(ByVal A() As Integer, ByVal n As Integer, ByVal lar As Integer, ByVal sml As Integer)

Console.WriteLine("The Array is :")

For i = 0 To n - 1

Console.WriteLine(" A(" & i & ") : " & A(i))

Next

Console.WriteLine("Largest Element is : " & lar)

Console.WriteLine("Smallest Element is : " & sml)

End Sub

Function largest(ByVal A() As Integer, ByVal n As Integer) As Integer

Dim lar, i As Integer

lar = A(0)

For i = 1 To n - 1

If A(i) > lar Then

lar = A(i)

End If

Next i

Return lar

End Function

Function smallest(ByVal A() As Integer, ByVal n As Integer) As Integer

Dim sml, i As Integer

sml = A(0)

For i = 1 To n - 1

If A(i) < sml Then

sml = A(i)

End If

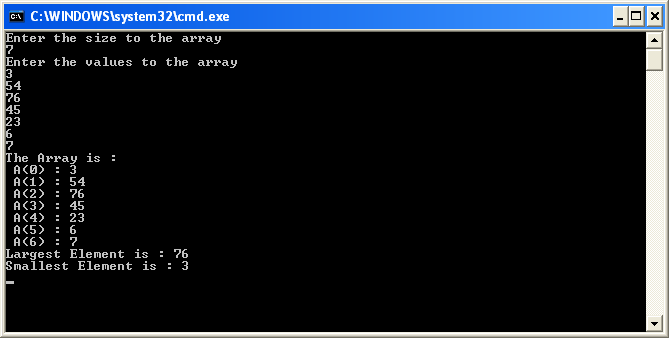
Next i

Return sml

End Function

End Module

**OUTPUT**



## Program to perform searching for an element in an array.

3. Write a Console Application in VB that will allow a user to perform searching for an element in an array. Display the values in the array, as well as, the result of the search. Your application should have a procedure which performs Binary Search. (Use subroutine for inputting values, displaying the array elements and function to perform binary search)

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim n, val, Array(100) As Integer

Dim result As Boolean

Console.WriteLine("Enter the size to the array")

n = Console.ReadLine()

Call Input(Array, n)

Call Display(Array, n)

Console.WriteLine("Enter the Element you want to search : ")

val = Console.ReadLine()

result = BinarySearch(Array, n, val)

If result Then

Console.WriteLine("Found")

Else

Console.WriteLine("Not Found")

End If

Console.Read()

End Sub

Sub Input(ByRef A() As Integer, ByVal n As Integer)

Console.WriteLine("Enter the values to the array")

For i = 0 To n - 1

A(i) = Console.ReadLine()

Next i

End Sub

Sub Display(ByVal A() As Integer, ByVal n As Integer)

Console.WriteLine("The Array is :")

For i = 0 To n - 1

Console.WriteLine(" A(" & i & ") : " & A(i))

Next

End Sub

Function BinarySearch(ByVal A() As Integer, ByVal n As Integer, ByVal val As Integer) As Boolean

Dim up, mid, low As Integer, flag As Boolean

low = 0

up = n

'While (low <> up)

' mid = (up + low) \ 2

' If A(mid) = val Then

' Return True

' ElseIf (A(mid) > val) Then

' up = mid - 1

' ElseIf A(mid) < val Then

' low = mid + 1

' End If

' If low = up Then

' Exit While

' End If

'End While

flag = False

Do While low <= up

mid = (low + up) / 2

If val = A(mid) Then

flag = True

Exit Do

ElseIf val < A(mid) Then

up = (mid - 1)

Else

low = (mid + 1)

End If

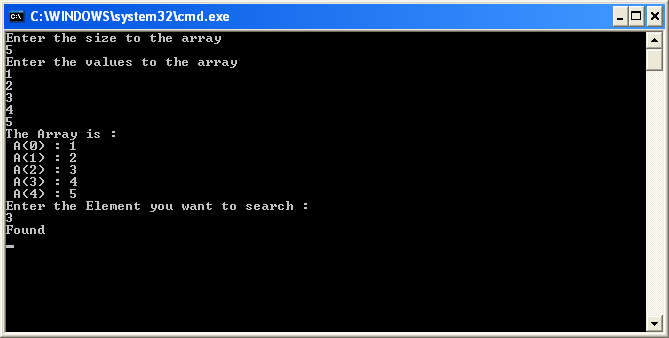
Loop

Return flag

End Function

End Module

**OUTPUT**



## 4. Program to store numeric values in an ‘m x n’ matrix and display the matrix and the sum of its diagonal elements.

4. Write a Console Application in VB to store numeric values in an ‘m x n’ matrix and display the matrix and the sum of its diagonal elements. (Use subroutine for inputting values and displaying the array elements, function to return the sum of matrix diagonal elements)

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim r As Integer, c, Matrix(100, 100) As Integer

Dim sumrd, sumld As Integer

Console.WriteLine("Enter the row of the Matrix")

r = Console.ReadLine()

Console.WriteLine("Enter the column of the Matrix")

c = Console.ReadLine()

ReDim Matrix(r, c)

Input(Matrix, r, c)

Display(Matrix, r, c)

sumrd = sumDiag(Matrix, r, c)

sumld = sumDiag2(Matrix, r, c)

Console.WriteLine("Sum of The Left Diagonal is " & sumld)

Console.WriteLine("Sum of The Right and Left Diagonals is " & sumrd)

Console.Read()

End Sub

Sub Input(ByRef A(,) As Integer, ByVal r As Integer, ByVal c As Integer)

Console.WriteLine("Enter the values to the array")

For i = 0 To r - 1

For j = 0 To c - 1

A(i, j) = Console.ReadLine()

Next j

Next i

End Sub

Sub Display(ByVal A(,) As Integer, ByVal r As Integer, ByVal c As Integer)

Console.WriteLine("The Matrix is :")

For i = 0 To r - 1

For j = 0 To c - 1

Console.Write(A(i, j) & vbTab)

Next j

Console.WriteLine()

Next

End Sub

Function sumDiag(ByVal Matrix(,) As Integer, ByVal r As Integer, ByVal c As Integer) As Integer

Dim sum As Integer = 0

For i = 0 To r - 1

For j = 0 To c - 1

If i + j = c - 1 Then

sum += Matrix(i, j)

End If

Next

Next

Return sum

End Function

Function sumDiag2(ByVal Matrix(,) As Integer, ByVal r As Integer, ByVal c As Integer) As Integer

Dim sum As Integer = 0

For i = 0 To r - 1

For j = 0 To c - 1

If i = j Then

sum += Matrix(i, j)

End If

Next

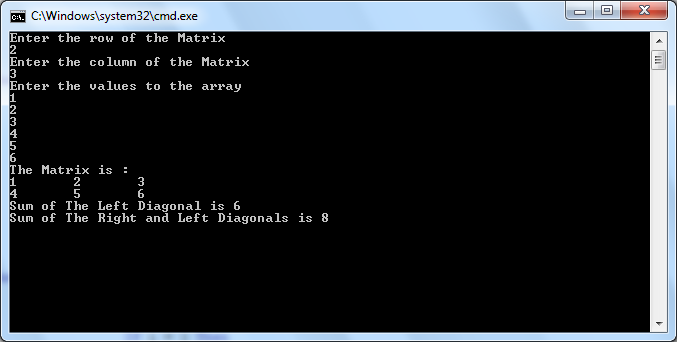
Next

Return sum

End Function

End Module

**OUTPUT**



## 5. Program to store numeric values in two ‘m x n’ matrix and display the two matrices along with their sum and difference

5. Write a Console Application in VB to store numeric values in two ‘m x n’ matrix. In the second form, display the two matrices along with their sum and difference. (Use subroutine for inputting values, displaying the array elements, to calculate the sum of two matrix into a third matrix and to calculate the difference of two matrix into a third matrix)

**Soln: Module1.vb**

Module Module1

Sub Main()

Dim r As Integer, c, Matrix(,), Matrix2(,) As Integer

Dim sum(,), diff(,) As Integer

Console.WriteLine("Enter the row of the Matrix")

r = Console.ReadLine()

Console.WriteLine("Enter the column of the Matrix")

c = Console.ReadLine()

ReDim Matrix(r, c)

ReDim Matrix2(r, c)

ReDim sum(r, c)

ReDim diff(r, c)

Console.WriteLine("Enter Into First Matrix/Array")

Input(Matrix, r, c)

Display(Matrix, r, c)

Console.WriteLine("Enter Into Second Matrix/Array")

Input(Matrix2, r, c)

Display(Matrix2, r, c)

Console.WriteLine("The Sum of the Matrices is :")

sum = sumMat(Matrix, Matrix2, r, c)

Display(sum, r, c)

Console.WriteLine("The Difference of the Matrices is :")

diff = diffMat(Matrix, Matrix2, r, c)

Display(diff, r, c)

Console.Read()

End Sub

Sub Input(ByRef A(,) As Integer, ByVal r As Integer, ByVal c As Integer)

Console.WriteLine("Enter the values to the array")

For i = 0 To r - 1

For j = 0 To c - 1

A(i, j) = Console.ReadLine()

Next j

Next i

End Sub

Sub Display(ByVal A(,) As Integer, ByVal r As Integer, ByVal c As Integer)

Console.WriteLine("The Matrix is :")

For i = 0 To r - 1

For j = 0 To c - 1

Console.Write(A(i, j) & vbTab)

Next j

Console.WriteLine()

Next

End Sub

Function sumMat(ByRef Matrix(,) As Integer, ByRef Matrix2(,) As Integer, ByVal r As Integer, ByVal c As Integer) As Integer(,)

Dim sum(r, c) As Integer

For i = 0 To r - 1

For j = 0 To c - 1

sum(i, j) = Matrix(i, j) + Matrix2(i, j)

Next

Next

Return sum

End Function

Function diffMat(ByRef Matrix(,) As Integer, ByRef Matrix2(,) As Integer, ByVal r As Integer, ByVal c As Integer) As Integer(,)

Dim sum(r, c) As Integer

For i = 0 To r - 1

For j = 0 To c - 1

sum(i, j) = Matrix(i, j) - Matrix2(i, j)

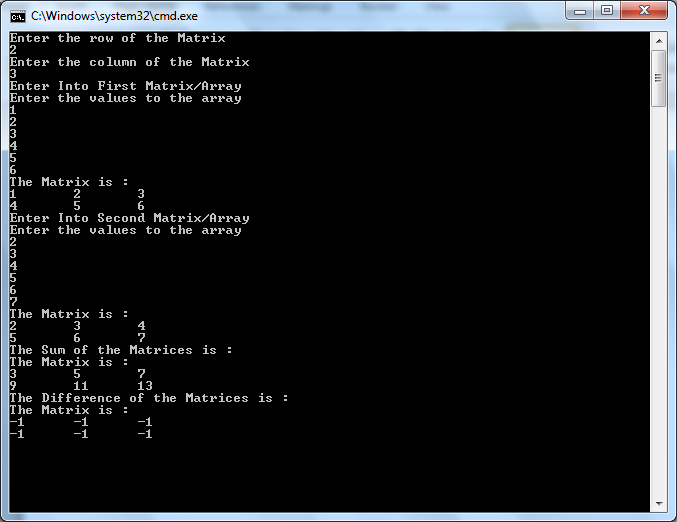
Next

Next

Return sum

End Function

End Module

**OUTPUT**  

# ARRAY GUI APPLICATIONS

## Program to Display the values in the array, the sum and average of all the elements in the array.

1. Using appropriate controls, design a Windows Forms Application that will take as input the number of numeric values in an array. Display the values in the array, as well as, the sum and average of all the elements in the array on the click of a button.

**Soln: Form1.vb**

Public Class Form1

Dim n As Integer, Arr(100) As Integer, i As Integer = 0

Private Sub btnOK\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOK.Click

n = txtSize.Text

btnOK.Enabled = False

txtSize.Enabled = False

btnNext.Enabled = True

txtNo.Enabled = True

txtNo.Focus()

End Sub

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

Dim sum, avg As Decimal

sum = 0

Label\_A.Text = "Enter Array(" & i + 1 & ")"

Arr(i) = txtNo.Text

i += 1

txtNo.Clear()

If i = n Then

btnNext.Enabled = False

txtNo.Enabled = False

For j = 0 To n - 1

txtArray.Text = txtArray.Text & " " & Arr(j)

sum = sum + Arr(j)

Next j

End If

avg = sum / n

txtSum.Text = sum

txtAvg.Text = avg

txtSum.Enabled = True

txtAvg.Enabled = True

txtNo.Focus()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

txtArray.Clear()

txtSize.Clear()

txtNo.Clear()

txtSum.Clear()

txtAvg.Clear()

txtSize.Enabled = True

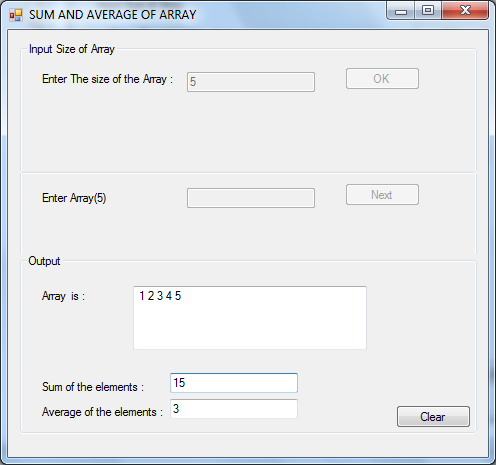
btnOK.Enabled = True

i = 0

End Sub

End Class

**OUTPUT**



## Program to Display the values in the array, along with the largest and smallest element in the array

2. Using appropriate controls, design a Windows Forms Application that will take as input the number of numeric values in an array. Display the values in the array, along with the largest and smallest element in the array on the click of a button.

**Soln: Form1.vb**

Public Class Form1

Dim n As Integer, Arr(100) As Integer, i As Integer = 0

Private Sub btnOK\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOK.Click

n = txtSize.Text

btnOK.Enabled = False

txtSize.Enabled = False

btnNext.Enabled = True

txtNo.Enabled = True

txtNo.Focus()

End Sub

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

Dim sml, lar As Decimal

sml = 0

Label\_A.Text = "Enter Array(" & i + 1 & ")"

Arr(i) = txtNo.Text

i += 1

txtNo.Clear()

If i = n Then

btnNext.Enabled = False

txtNo.Enabled = False

sml = Arr(0)

lar = Arr(0)

txtArray.Text = Arr(0)

For j = 1 To n - 1

'Finding elements

If Arr(j) < sml Then

sml = Arr(j)

End If

If Arr(j) > lar Then

lar = Arr(j)

End If

'Display

txtArray.Text = txtArray.Text & " " & Arr(j)

Next j

End If

txtSml.Text = sml

txtLar.Text = lar

txtSml.Enabled = True

txtLar.Enabled = True

txtNo.Focus()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

txtArray.Clear()

txtSize.Clear()

txtNo.Clear()

txtSize.Enabled = True

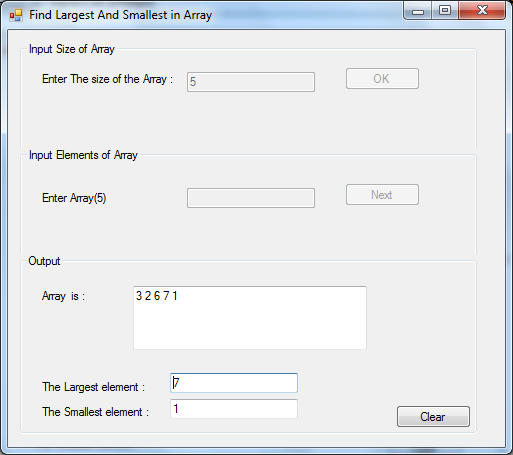
btnOK.Enabled = True

i = 0

End Sub

End Class

**OUTPUT**



## Program to Display the values in the array, as well as, the result of the search

3. Using appropriate controls, design a Windows Forms Application that will allow a user to perform searching for an element in an array. Display the values in the array, as well as, the result of the search, on the click of a button. Your application should have a procedure which performs Binary Search.

**Soln : Form1.vb**

Public Class Form1

Dim n As Integer, Arr(100) As Integer, i As Integer = 0

Private Sub btnOK\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOK.Click

n = txtSize.Text

btnOK.Enabled = False

txtSize.Enabled = False

btnNext.Enabled = True

txtNo.Enabled = True

txtNo.Focus()

End Sub

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

Dim sml As Decimal

sml = 0

Label\_A.Text = "Enter Array(" & i + 1 & ")"

Arr(i) = txtNo.Text

i += 1

txtSrch.Enabled = False

btnFind.Enabled = False

txtNo.Focus()

If i = n Then

txtNo.Enabled = False

btnNext.Enabled = False

txtSrch.Enabled = True

txtSrch.Focus()

btnFind.Enabled = True

End If

txtNo.Clear()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

txtArray.Clear()

txtSize.Clear()

txtNo.Clear()

txtFound.Clear()

txtSrch.Clear()

txtSize.Enabled = True

btnOK.Enabled = True

i = 0

End Sub

Private Sub btnFind\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnFind.Click

Dim element As Integer

Dim result As Boolean

element = txtSrch.Text

result = BinarySearch(Arr, n, element)

txtArray.Text = Arr(0)

For j = 1 To n - 1

'Display

txtArray.Text = txtArray.Text & " " & Arr(j)

Next j

txtFound.Enabled = True

If result Then

txtFound.Text = "Found"

Else

txtFound.Text = "Not Found"

'txtFound.Text = result

End If

End Sub

Function BinarySearch(ByVal A() As Integer, ByVal n As Integer, ByVal val As Integer) As Boolean

Dim up, mid, low As Integer, flag As Boolean

low = 0

up = n

'While (low <> up)

' mid = (up + low) \ 2

' If A(mid) = val Then

' Return True

' ElseIf (A(mid) > val) Then

' up = mid - 1

' ElseIf A(mid) < val Then

' low = mid + 1

' End If

' If low = up Then

' Exit While

' End If

'End While

'Return False

flag = False

Do While low <= up

mid = (low + up) / 2

If val = A(mid) Then

flag = True

Exit Do

ElseIf val < A(mid) Then

up = (mid - 1)

Else

low = (mid + 1)

End If

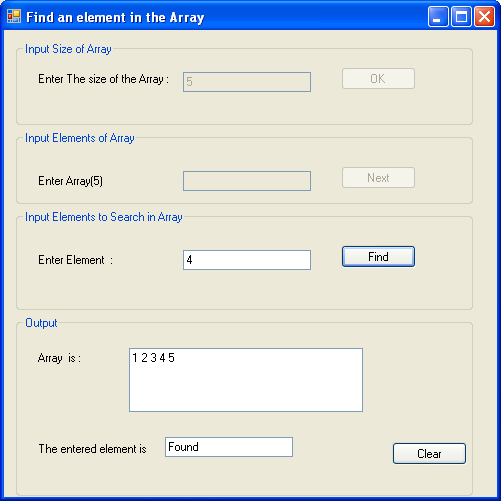
Loop

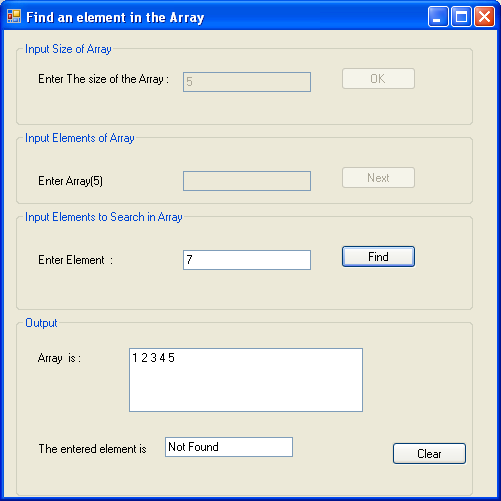
Return flag

End Function

End Class

**OUTPUT**





## Program to store numeric values in an ‘m x n’ matrix and display the matrix and the sum of its diagonal elements

4. Using appropriate controls, design a Windows Forms Application to store numeric values in an ‘m x n’ matrix and display the matrix and the sum of its diagonal elements, on the click of a button.

**Soln: Form1.vb**

Public Class Form1

Dim Matrix(,) As Integer

Dim r, c, k As Integer

Private Sub btnOK1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOK1.Click

If txtC.Text <> "" And txtR.Text <> "" Then

r = txtR.Text

c = txtC.Text

txtC.Enabled = False

txtR.Enabled = False

btnOK1.Enabled = False

txtIn.Enabled = True

ReDim Matrix(r - 1, c - 1)

k = 0

End If

End Sub

Private Sub btnOK2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOK2.Click

For i = 0 To r - 1

For j = 0 To c - 1

txtOut.Text = txtOut.Text & " " & Matrix(i, j)

Next

txtOut.Text = txtOut.Text & vbCrLf

Next

txtOut.Enabled = True

btnOK2.Enabled = False

txtSumDia.Enabled = True

txtSumDia.Text = SumDiag()

End Sub

Private Sub txtIn\_TextChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles txtIn.TextChanged

If txtIn.Lines.Length = c + 1 Then

For j = 0 To c - 1

Matrix(k, j) = CInt(txtIn.Lines(j))

Next

If k < r - 1 Then

k += 1

txtIn.Clear()

Else

txtIn.Clear()

txtIn.Enabled = False

btnOK2.Enabled = True

End If

End If

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

txtOut.Enabled = False

txtOut.Clear()

txtR.Enabled = True

txtC.Enabled = True

btnOK1.Enabled = True

txtSumDia.Clear()

txtSumDia.Enabled = False

End Sub

Public Function SumDiag() As Integer

Dim sum As Integer = 0

For i = 0 To r - 1

For j = 0 To c - 1

If i + j = c - 1 Then

sum += Matrix(i, j)

End If

Next

Next

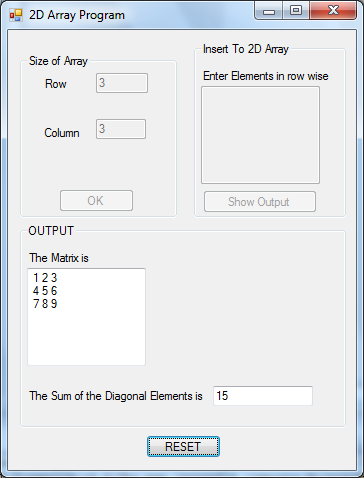
Return sum

End Function

End Class

# 

**OUTPUT**



# Dynamic Forms Worksheet

## Application on Quadratic Equation and display its roots

1. Design a GUI Application to take as input the co-efficient of a quadratic equation Ax+Bx2+C. Add a second form in the application and, create the interface dynamically to display roots, if found or display a message if roots are imaginary, also in the second form. The second form should behave as a dialog box.

**Soln: Form1.vb**

Public Class Form1

Public root, root2 As Integer

Public flag As Integer = 0

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Dim a, b, c, D As Integer

a = TextBox1.Text

b = TextBox2.Text

c = TextBox3.Text

D = (b \* b) - (4 \* a \* c)

If D = 0 Then

'roots equal

root = ((-b) ^ 2 + Math.Sqrt(b ^ 2 - 4 \* a \* c)) / (2 \* a)

root2 = root

flag = 0

FrmOutput.Show()

ElseIf D > 0 Then

'real

root = ((-b) ^ 2 + Math.Sqrt(b ^ 2 - 4 \* a \* c)) / (2 \* a)

root2 = ((-b) ^ 2 - Math.Sqrt(b ^ 2 - 4 \* a \* c)) / (2 \* a)

flag = 1

FrmOutput.Show()

Else

'img

flag = 2

FrmOutput.ShowDialog()

End If

End Sub

End Class

**FrmOutput.vb**

Public Class FrmOutput

Private Sub FrmOutput\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim label As New System.Windows.Forms.Label

Dim txtbox1 As New System.Windows.Forms.TextBox

Dim txtbox2 As New System.Windows.Forms.TextBox

If Form1.flag = 0 Then

With label

.Text = "The Roots are equal"

.Left = 30

.Top = 30

End With

Me.Controls.Add(label)

Me.Text = "Equal Roots"

With txtbox1

.Text = Form1.root

.Top = 70

.Left = 30

End With

With txtbox2

.Text = Form1.root2

.Top = 90

.Left = 30

End With

Me.Controls.Add(txtbox1)

Me.Controls.Add(txtbox2)

ElseIf Form1.flag = 1 Then

With label

.Text = "The Roots are Real :"

.Left = 30

.Top = 30

End With

Me.Controls.Add(label)

Me.Text = "Real Roots"

With txtbox1

.Text = Form1.root

.Top = 70

.Left = 30

End With

With txtbox2

.Text = Form1.root2

.Top = 90

.Left = 30

End With

Me.Controls.Add(txtbox1)

Me.Controls.Add(txtbox2)

Else

With label

.Text = "Imaginary roots"

.Left = 20

.Top = 30

End With

Me.Controls.Add(label)

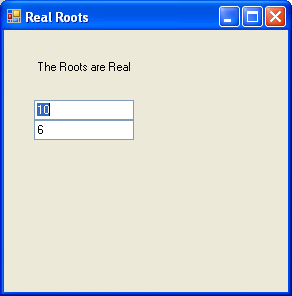
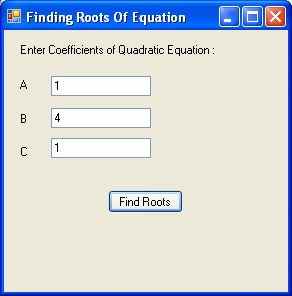
Me.Text = "Imaginary Roots"

End If

End Sub

End Class

**OUTPUT**



# Class Worksheet

## Windows Application that uses methods of the class Triangle to find the area of a triangle.

1. Design a Windows Application that uses methods of the class **Triangle** to find the area of a triangle. The class has a properties base and height. Provide property procedures to set & get the value of its properties, a method to calculate the area and three constructors (The default constructor, a constructor to assign equal value to base and height and a constructor to assign different values to base and height). Test this class by designing the GUI for user to input base and height. Have a CheckBox Control where user can specify that value is same. Create the object accordingly based on the specified input. If no values were provided, then create the default object if user clicks to view its area. This process should continue for any number of inputs the user chooses to test, therefore provide a reset or clear button.

**Soln: Form1.vb**

Public Class Form1

Dim base, hght As Integer

Dim Tri As New Triangle

Dim findarea As Single

Private Sub CheckBox1\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged

If CheckBox1.Checked = True Then

TextBox2.Enabled = False

Else

TextBox2.Enabled = True

End If

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Label4.Text = ""

If TextBox1.Text <> "" Then

base = TextBox1.Text

If CheckBox1.Checked = True Then

hght = base

Else

If TextBox2.Text <> "" Then

hght = TextBox2.Text

Else

hght = 1

End If

End If

Tri.b = base

Tri.h = hght

findarea = Tri.area()

Else

findarea = Tri.area()

End If

Label4.Text = "Area is: " & " " & findarea

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

TextBox1.Clear()

TextBox2.Clear()

End Sub

End Class

**Triangle.vb**

Public Class Triangle

Private base As Integer

Private height As Integer

Public Sub New()

base = height = 1

End Sub

Public Sub New(ByVal x As Integer)

base = height = x

End Sub

Public Sub New(ByVal b As Integer, ByVal h As Integer)

base = b

height = h

End Sub

Property b() As Integer

Get

b = base

End Get

Set(ByVal value As Integer)

base = value

End Set

End Property

Property h() As Integer

Get

h = height

End Get

Set(ByVal value As Integer)

height = value

End Set

End Property

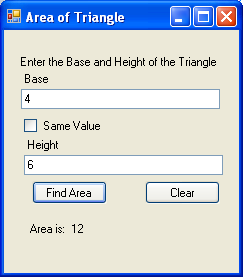
Function area() As Single

Return (0.5 \* base \* height)

End Function

End Class

**OUTPUT**



## Windows Application that uses methods of the class to find the number of digits, sum of the digits of a number and reverse of the number.

1. Design a Windows Application that uses methods of the class **Number [Use a Class Library to design the class]** to find the number of digits, sum of the digits of a number and reverse of the number. The class has a property called *magnitude* with a default value of 1, which stores the value entered. Provide property procedure to set & get the value of *magnitude* and methods to perform the above mentioned tasked. Test the class such that a number is taken as input and output should be the number of digits, sum of its digits and the reverse of the number.

**Soln: numberlibrary\Number.vb**

Public Class Number

Private magnitude As Integer

Public Sub New()

magnitude = 1

End Sub

Public Sub New(ByVal n As Integer)

magnitude = n

End Sub

Public Property data() As Integer

Get

Return magnitude

End Get

Set(ByVal value As Integer)

magnitude = value

End Set

End Property

Function sumDigits() As Integer

Dim sum, tn, r As Integer

sum = 0

tn = magnitude

While tn > 0

r = tn Mod 10

sum = sum + r

tn /= 10

End While

Return sum

End Function

Function noOfDigits() As Integer

Dim count, tn As Integer

count = 0

tn = magnitude

While tn > 0

count = count + 1

tn /= 10

End While

Return count

End Function

Function reverse() As Integer

Dim rev, tn, r As Integer

rev = 0

tn = magnitude

While tn > 0

r = tn Mod 10

rev = rev \* 10 + r

tn /= 10

End While

Return rev

End Function

End Class

**Form1.vb**

Imports System.Text.RegularExpressions

Public Class Form1

Dim n As Integer

Private Sub btnok\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnrev.Click

If txtnum.Text <> "" And Regex.IsMatch(txtnum.Text, "[^a-zA-Z]") = True Then

n = CInt(txtnum.Text)

Dim obj As New numberlibrary.Number(n)

txt.Text = obj.reverse()

End If

End Sub

Private Sub btnclear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnclear.Click

txtnum.Clear()

txt.Clear()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSum.Click

If txtnum.Text <> "" And Regex.IsMatch(txtnum.Text, "[^a-zA-Z]") = True Then

n = CInt(txtnum.Text)

Dim obj As New numberlibrary.Number(n)

txt.Text = obj.sumDigits()

End If

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNo.Click

If txtnum.Text <> "" And Regex.IsMatch(txtnum.Text, "[^a-zA-Z]") = True Then

n = CInt(txtnum.Text)

Dim obj As New numberlibrary.Number(n)

txt.Text = obj.noOfDigits()

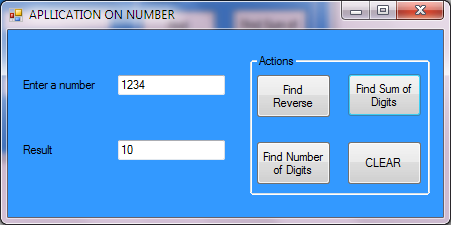
End If

End Sub

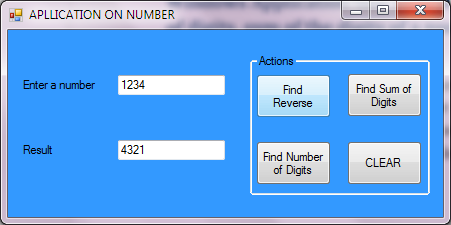
End Class

**OUTPUT**

**Find sum of the digits**



**Find Reverse of a number**



# Class Worksheet on Strings

## Windows Application that uses methods of a class to determine if an inputted word is a palindrome.

1. Design a Windows Application that uses methods of a class to determine if an inputted word is a palindrome. The class has a property called *word* with the word “default string” as default. Provide property procedure to set & get the value of *word* and a method to return Boolean value for determining if word is Palindrome. [Using Class Library]

**Soln: Palindromelib\Palindrome.vb**

Public Class Palindrome

Private word As String

Public Sub New()

word = "empty"

End Sub

Public Property wordvalue() As String

Get

Return word

End Get

Set(ByVal value As String)

word = value

End Set

End Property

Public Function ispalinddrome() As Boolean

Dim i As Integer

Dim reverse As String

reverse = ""

reverse = StrReverse(word)

If (String.Compare(word, reverse) = 0) Then

Return True

Else

Return False

End If

End Function

End Class

**Form1.vb**

Public Class Form1

Dim obj As New palindromelib.palindrome

Private Sub btnok\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnok.Click

obj.wordvalue = CStr(txtinput.Text)

If obj.ispalinddrome() = True Then

txtoutput.Text = "A Palindrome"

Else

txtoutput.Text = "Not a Palindrome"

End If

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

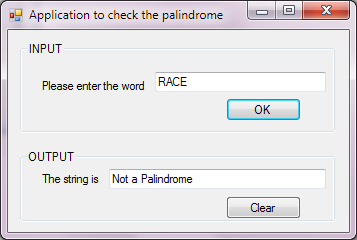
txtinput.Clear()

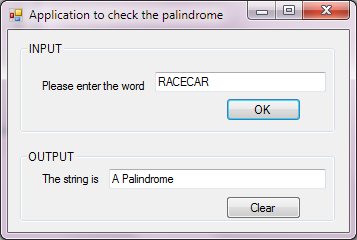
txtoutput.Clear()

End Sub

End Class

**OUTPUT**

****

****

## Windows Application that uses methods of a class to display the number of words, number of vowels, number of digits and number of special symbols present in a sentence.

1. Design a Windows Application that uses methods of a class to display the number of words, number of vowels, number of digits and number of special symbols present in a sentence. The class has a property called *sentence* with the word “default string” as default. Provide property procedure to set & get the value of *sentence* and methods to perform the above mentioned tasks. [Using Class Library]

**Soln: Findingword\words.vb**

Imports System.Text.RegularExpressions

Public Class words

Private sentence As String

Dim length As Integer

Public Sub New()

sentence = "empty"

End Sub

Public Property sentencevalue() As String

Get

Return sentence

End Get

Set(ByVal value As String)

sentence = value

End Set

End Property

Public Function getword() As String()

Dim i, j As Integer

Dim word As String

Dim arr() As String

j = 0

ReDim arr(j)

sentence = sentence.Trim() & " "

For i = 0 To sentence.Length - 1

word = ""

'count Words

While sentence(i) <> " "

' word = word & sentence(i)

word = word & sentence(i)

i = i + 1

length = i

End While

arr(j) = word

j = j + 1

ReDim Preserve arr(j)

Next

Return arr

End Function

Public Function getdigits() As Integer

Dim i, noOfDigits As Integer

Dim word As String

noOfDigits = 0

word = sentence

For i = 0 To length

If sentence(i) = "1" Or sentence(i) = "2" Or sentence(i) = "3" Or sentence(i) = "4" Or sentence(i) = "5" Or sentence(i) = "6" Or sentence(i) = "7" Or sentence(i) = "8" Or sentence(i) = "9" Or sentence(i) = "0" Then

noOfDigits = noOfDigits + 1

End If

Next i

Return noOfDigits

End Function

Public Function getvowels() As Integer

Dim i, noOfvowels As Integer

Dim word As String

noOfvowels = 0

word = sentence

Console.WriteLine("length I s" & length)

For i = 0 To length

If sentence(i) = "A" Or sentence(i) = "a" Or sentence(i) = "E" Or sentence(i) = "e" Or sentence(i) = "i" Or sentence(i) = "I" Or sentence(i) = "O" Or sentence(i) = "o" Or sentence(i) = "U" Or sentence(i) = "u" Then

noOfvowels = noOfvowels + 1

End If

Next i

Console.WriteLine("no " & noOfvowels)

Return noOfvowels

End Function

Public Function getchars() As Integer

Dim noOfsc As Integer

Dim word As String

noOfsc = 0

word = sentence

For i = 0 To length

If Regex.IsMatch(word(i), "[^a-zA-Z0-9 ]") Then

noOfsc = noOfsc + 1

End If

Next i

'Console.WriteLine("no " & noOfvowels)

Return noOfsc

End Function

End Class

**Form1.vb**

Public Class Form1

Dim obj As New findingwords.words

Private Sub btnfindword\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnfindword.Click

Dim array() As String

Dim i, digits, vowels, spc As Integer

digits = 0

vowels = 0

spc = 0

If txtsentence.Text <> "" Then

obj.sentencevalue = CStr(txtsentence.Text)

lstwords.Items.Clear()

array = obj.getword()

For i = 0 To UBound(array) - 1

lstwords.Items.Add(array(i))

Next

digits = obj.getdigits()

vowels = obj.getvowels()

spc = obj.getchars()

lstwords.Items.Add("Number of Words is " & UBound(array))

lstwords.Items.Add("Number of Digits is " & digits)

lstwords.Items.Add("Number of Vowels is " & vowels)

lstwords.Items.Add("Number of Special Characters is " & spc)

End If

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

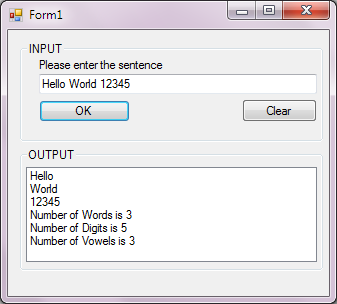
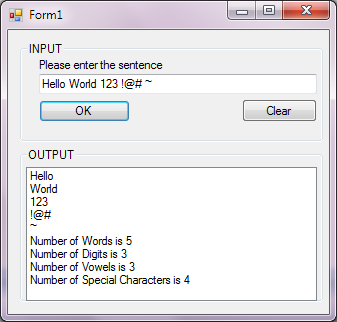
txtsentence.Clear()

lstwords.Items.Clear()

End Sub

End Class

**OUTPUT**

## Windows Application that uses methods of a class to display the longest and the shortest word in a sentence.

1. Design a Windows Application that uses methods of a class to display the longest and the shortest word in a sentence. The class has a property called *sentence* with the word “default string” as default. Provide property procedure to set & get the value of *sentence* and methods to perform the above mentioned tasks.

**Soln: LongShortlib\LongShortlib.vb**

Public Class LongShort

Private sentence As String

Public Sub New()

sentence = "default string"

End Sub

Public Property getvalue()

Get

Return sentence

End Get

Set(ByVal value)

sentence = value

End Set

End Property

Public Function getlongestword() As String

Dim i, len As Integer

Dim word, rword As String

len = 0

sentence = sentence.Trim & " "

word = ""

rword = ""

For i = 0 To sentence.Length - 1

word = ""

While sentence(i) <> " "

word = word & sentence(i)

i = i + 1

End While

If word.Length > len Then

len = word.Length

rword = word

End If

Next

Return rword

End Function

Public Function getshortestword() As String

Dim i, len As Integer

Dim word, rword As String

len = 0

word = ""

rword = ""

sentence = sentence.Trim & " "

For i = 0 To sentence.Length - 1

word = ""

While sentence(i) <> " "

word = word & sentence(i)

i = i + 1

End While

If word.Length < len Or len = 0 Then

len = word.Length

rword = word

End If

Next

Return rword

End Function

End Class

**Q4LongShortWord\Form1.vb**

Public Class Form1

Dim Sentence As New LongShortlib.LongShort

Private Sub btnFind\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnFind.Click

Sentence.getvalue = txtin.Text

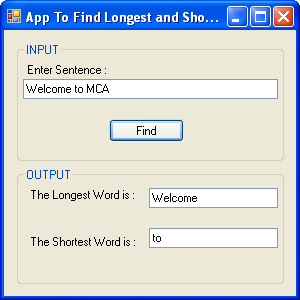
txtShort.Text = Sentence.getshortestword

txtLong.Text = Sentence.getlongestword

End Sub

End Class

**OUTPUT**



## Windows Application that uses methods of a class to display whether the words entered by the user are Anagrams of each other.

1. Design a Windows Application that uses methods of a class to display whether the words entered by the user are Anagrams of each other. The class has one property called *word* with the word “default” as default. Provide property procedure to set & get the value of *word* and method to perform the above mentioned tasks. A word is an anagram of another word, if the same letters where used to form the words. Example: silent and listen are anagrams. “*Each word is to be treated as an object of the class”*.

**Soln: Form1.vb**

Public Class Form1

Dim word1, word2 As New Word

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click, Button4.Click

If TextBox1.Text <> "" And TextBox2.Text <> "" Then

word1.data = TextBox1.Text

word2.data = TextBox2.Text

If Word.isAnagram(word1, word2)= True Then

result.Text = "The Words form an anagram"

Else

result.Text = "The Words do not form an anagram"

End If

End If

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click, Button3.Click

TextBox2.Clear()

TextBox1.Clear()

result.Text = ""

End Sub

End Class

**Word.vb**

Public Class Word

Dim myWord As String

Public Sub New()

myWord = "default"

End Sub

Public Sub New(ByVal n As String)

myWord = n

End Sub

Public Property data() As String

Get

Return myWord

End Get

Set(ByVal value As String)

myWord = value

End Set

End Property

Shared Function isAnagram(ByRef word1 As Word, ByRef word2 As Word) As Boolean

word1.myWord = word1.myWord.ToLower()

word2.myWord = word2.myWord.ToLower()

If word1.myWord.Length <> word2.myWord.Length Then

Return False

ElseIf word1.myWord.Length = word2.myWord.Length Then

word1.myWord = word1.sortWord(word1.myWord)

word2.myWord = word2.sortWord(word2.myWord)

Form1.test.Text = word1.myWord

Form1.test.Text = Form1.test.Text & word2.myWord

Dim i, j As Integer

i = 0 : j = 0

While i < word1.myWord.Length

If word1.myWord(i) = word2.myWord(i) Then

j += 1

End If

i += 1

End While

If j = word1.myWord.Length Then

Return True

Else

Return False

End If

End If

End Function

Function sortWord(ByRef str As String) As String

Dim len As Integer

Dim newWord As String

len = str.Length

newWord = "a"

'Console.WriteLine("Length is " & len)

Dim char1() As Char

char1 = str.ToCharArray()

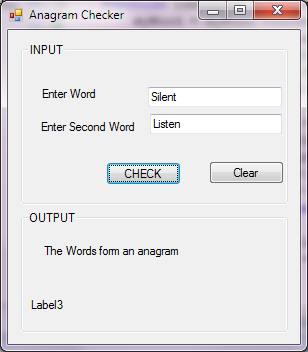
System.Array.Sort(char1)

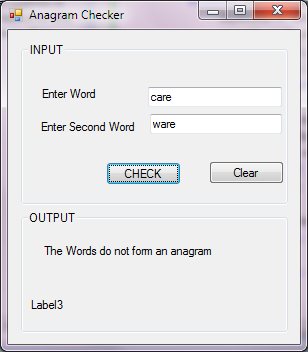
Return char1

End Function

End Class

**OUTPUT**





# Numeric Class Worksheet

## Windows Application that uses methods of the class Armstrong to determine if a number is an Armstrong number or not.

1. Design a Windows Application that uses methods of the class **Armstrong** to determine if a number is an Armstrong number or not. The class has a property called *number* with a default value of 1. Provide property procedure to set & get the value of *number* and a method to return Boolean value for whether number is Armstrong or not. Test the class. **[The Class Armstrong can be in the same application]**

**Soln: Armstrong.vb**

1. Public Class Armstrong
2. Dim number As Integer
3. Public Sub New()
4. number = 1
5. End Sub
6. Public Property data() As Integer
7. Get
8. Return number
9. End Get
10. Set(ByVal value As Integer)
11. number = value
12. End Set
13. End Property
14. Public Function Check() As Boolean
15. Dim rev, tn, r As Integer
16. rev = 0
17. tn = number
18. While tn > 0
19. r = tn Mod 10
20. rev = rev + r ^ 3
21. tn = tn \ 10
22. 'Console.WriteLine(" r n " & n)
23. End While
24. ' Console.WriteLine("Reverse no: " & rev)
25. If rev = number Then
26. ' Console.WriteLine(" The Number is an Armstrong Number ")
27. Return True
28. Else
29. 'Console.WriteLine("The Number is not an Armstrong Number")
30. Return False
31. End If
32. End Function
33. End Class

**Form1.vb**

Public Class Form1

Dim Arm As New Armstrong

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Arm.data = TextBox1.Text

Label3.Text = ""

Label3.Text = "Result:"

If Arm.Check() Then

Label3.Text = Label3.Text & Arm.data & " is an Armstrong Number"

Else

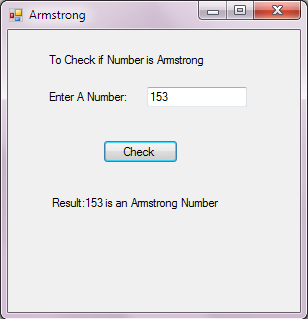
Label3.Text = Label3.Text & Arm.data & " is not an Armstrong Number"

End If

End Sub

End Class

**OUTPUT**

****

## Windows Application that uses methods of the class Factors to find the factors of a number

4. Design a Windows Application that uses methods of the class **Factors** to find the factors of a number and determine if number is a Prime or Non-prime and, Abundant or Perfect by using methods of the class. The class has two properties numberand an array to store its factors. A default constructor for initializing number to 1 and array accordingly i.e. array should contain the factor of 1. Property procedure to set and get value of number. While setting the number, the array should also be reset to contain factors of the new number. A property procedure for getting values of the array. Methods to determine prime, abundant and perfect, taking the help of the factors array accordingly. **[Use Class Library to design the class Number]**

**Soln:**

**Factorlib\factor.vb**

Public Class Factors

Private n, Counter As Integer

Private Array() As Integer

Public Sub New()

n = 1

ReDim array(0)

Array(0) = 1

End Sub

Public Property data() As Integer

Get

Return n

End Get

Set(ByVal value As Integer)

n = value

Dim j As Integer = 0

For i As Integer = 1 To n

If n Mod i = 0 Then

ReDim Preserve Array(j)

Array(j) = i

Counter = j

j = j + 1

End If

Next

End Set

End Property

Public ReadOnly Property count() As Integer

Get

Return Counter

End Get

End Property

ReadOnly Property getarray() As Integer()

Get

Return Array

End Get

End Property

Function Abundant() As Boolean

Dim i, sum As Integer

i = 0

sum = 0

While i < Counter

sum += Array(i)

i += 1

End While

If sum > n Then

Return True

Else

Return False

End If

End Function

Function PerfectNo() As Boolean

Dim i, sum As Integer

i = 0

sum = 0

While i < Counter

sum += Array(i)

i += 1

End While

If sum = n Then

Return True

Else

Return False

End If

End Function

Function isPrime() As Boolean

Dim i As Integer, Flag As Boolean

Flag = True

If n <> 2 Then

For i = 2 To n / 2

If n Mod i = 0 Then

Flag = False

End If

If Flag = False Then

Exit For

End If

'Console.WriteLine("i " & i)

Next

End If

If Flag = False Then

'Console.WriteLine("It is not a Prime Number")

Return False

Else

'Console.WriteLine("It is a Prime Number")

Return True

End If

End Function

End Class

**Form1.vb**

Public Class Form1

Dim No As New factorlib.factor

Dim Array() As Integer

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

If TextBox1.Text <> "" Then

No.data = TextBox1.Text

Array = No.getarray

TextBox2.Clear()

For i = 0 To No.count

TextBox2.Text = TextBox2.Text & Array(i) & " "

Next

TextBox1.Focus()

If No.isPrime() = True Then

lblPrime.Text = "It is a Prime Number"

Else

lblPrime.Text = "It is not a Prime Number"

End If

If No.Abundant() = True Then

LblANo.Text = "It is an Abundant Number"

Else

LblANo.Text = "It is Not an Abundant Number!"

End If

If No.PerfectNo() = True Then

lblPNo.Text = "It is a Perfect Number"

Else

lblPNo.Text = "It is Not a Perfect Number!"

End If

End If

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

TextBox1.Clear()

TextBox2.Clear()

LblANo.Text = ""

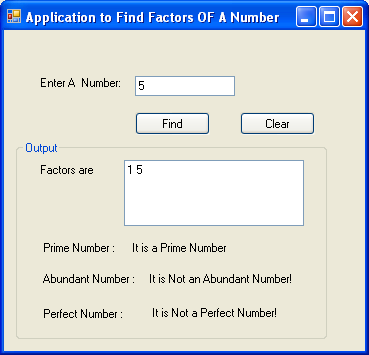
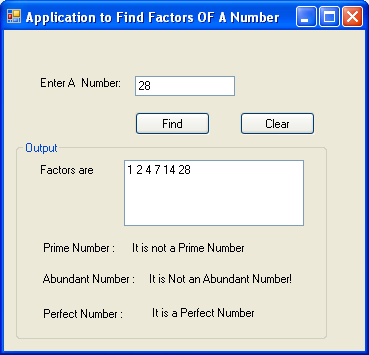
lblPNo.Text = ""

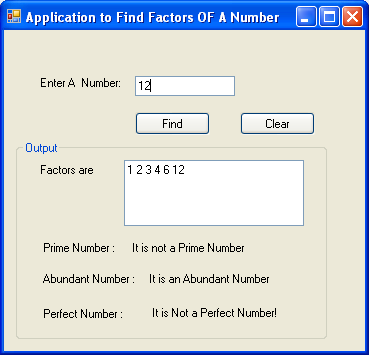
lblPrime.Text = ""

End Sub

End Class

**OUTPUT**



# Mini Projects

## Calculator

**Soln: Form1.vb**

Public Class Form1

Dim value As Decimal = 0

Dim val2 As Decimal = 0

Dim ans As Decimal = 0

Public Ns As String = "Dec"

Public New\_Ns As String = "Dec"

Private Sub btn1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn1.Click

Textarea.Text = Textarea.Text & 1

End Sub

Private Sub btn0\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn0.Click

Textarea.Text = Textarea.Text & 0

End Sub

Private Sub btn2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn2.Click

Textarea.Text = Textarea.Text & 2

End Sub

Private Sub btn3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn3.Click

Textarea.Text = Textarea.Text & 3

End Sub

Private Sub btn4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn4.Click

Textarea.Text = Textarea.Text & 4

End Sub

Private Sub btn5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn5.Click

Textarea.Text = Textarea.Text & 5

End Sub

Private Sub btn6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn6.Click

Textarea.Text = Textarea.Text & 6

End Sub

Private Sub btn7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn7.Click

Textarea.Text = Textarea.Text & 7

End Sub

Private Sub btn8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn8.Click

Textarea.Text = Textarea.Text & 8

End Sub

Private Sub btn9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btn9.Click

Textarea.Text = Textarea.Text & 9

End Sub

Private Sub btnClear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click

Textarea.Clear()

value = 0

val2 = 0

ans = 0

TextBox1.Clear()

End Sub

Private Sub btneq\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btneq.Click

If Textarea.Text <> "" Then

val2 = Textarea.Text

Textarea.Clear()

If TextBox1.Text.Contains("+") Then

ans = value + val2

TextBox1.Text = TextBox1.Text & val2

ElseIf TextBox1.Text.Contains("-") Then

ans = value - val2

TextBox1.Text = TextBox1.Text & val2

ElseIf TextBox1.Text.Contains("/") Then

ans = value / val2

TextBox1.Text = TextBox1.Text & val2

ElseIf TextBox1.Text.Contains("\*") Then

ans = value \* val2

TextBox1.Text = TextBox1.Text & val2

ElseIf TextBox1.Text.Contains("MOD") Then

ans = value Mod val2

TextBox1.Text = TextBox1.Text & val2

ElseIf TextBox1.Text.Contains("AND") Then

ans = value And val2

TextBox1.Text = TextBox1.Text & val2

ElseIf TextBox1.Text.Contains("OR") Then

ans = value Or val2

TextBox1.Text = TextBox1.Text & val2

ElseIf TextBox1.Text.Contains("XOR") Then

ans = value Xor val2

TextBox1.Text = TextBox1.Text & val2

ElseIf TextBox1.Text.Contains("^") Then

ans = Math.Pow(value, val2)

End If

Textarea.Text = ans

value = 0

'val2 = 0

End If

End Sub

Private Sub StandardToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles StandardToolStripMenuItem.Click

GroupBox2.Visible = False

End Sub

Private Sub btnadd\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnadd.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

TextBox1.Text = Textarea.Text & "+"

'Textarea.Text = Textarea.Text

Else

value += Textarea.Text

'TextBox1.Text = TextBox1.Text & Textarea.Text

TextBox1.Text = TextBox1.Text & Textarea.Text & "+"

End If

Textarea.Clear()

End If

End Sub

Private Sub btndiff\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndiff.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

TextBox1.Text = Textarea.Text & "-"

Else

value -= Textarea.Text

TextBox1.Text = TextBox1.Text & Textarea.Text & "-"

End If

Textarea.Clear()

End If

End Sub

Private Sub btndiv\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndiv.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

TextBox1.Text = Textarea.Text & "/"

Else

value /= Textarea.Text

TextBox1.Text = TextBox1.Text & Textarea.Text & "/"

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

End If

End Sub

Private Sub btnmul\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnmul.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

TextBox1.Text = Textarea.Text & "\*"

Else

value \*= Textarea.Text

TextBox1.Text = TextBox1.Text & Textarea.Text & "\*"

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

End If

End Sub

Private Sub ScientificToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ScientificToolStripMenuItem.Click

GroupBox2.Visible = True

End Sub

Private Sub ExitToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click

Me.Close()

End Sub

Private Sub btnSign\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSign.Click

Dim check As Decimal

If Textarea.Text <> "" Then

check = Textarea.Text

check = check \* -1

Textarea.Text = check

Else

Textarea.Text = "-"

End If

End Sub

Private Sub btnSin\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSin.Click

If Textarea.Text <> "" Then

ans = Math.Sin((Textarea.Text \* Math.PI) / 180)

Else

ans = 0

End If

Textarea.Clear()

Textarea.Text = ans

End Sub

Private Sub btnCos\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCos.Click

If Textarea.Text <> "" Then

ans = Math.Cos(Textarea.Text)

Else

ans = 0

End If

Textarea.Clear()

Textarea.Text = ans

End Sub

Private Sub btnLog\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnLog.Click

If Textarea.Text <> "" Then

ans = Math.Log(Textarea.Text)

Else

ans = 0

End If

Textarea.Clear()

Textarea.Text = ans

End Sub

Private Sub Button19\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button19.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

TextBox1.Text = Textarea.Text & " MOD "

Else

value = value Mod Textarea.Text

TextBox1.Text = TextBox1.Text & Textarea.Text & " MOD "

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

End If

End Sub

Private Sub Button21\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button21.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

TextBox1.Text = Textarea.Text & " AND "

Else

value = value And Textarea.Text

TextBox1.Text = TextBox1.Text & Textarea.Text & " AND "

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

End If

End Sub

Private Sub Button23\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button23.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

TextBox1.Text = Textarea.Text & " OR "

Else

value = value Or Textarea.Text

TextBox1.Text = TextBox1.Text & Textarea.Text & " OR "

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

End If

End Sub

Private Sub Button25\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button25.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Not Textarea.Text

TextBox1.Text = " NOT " & Textarea.Text

Else

value = 0

value = Not Textarea.Text

TextBox1.Text = TextBox1.Text & Textarea.Text & " NOT "

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

Textarea.Text = value

End If

End Sub

Private Sub Button26\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button26.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

TextBox1.Text = Textarea.Text & " XOR "

Else

value = value Xor Textarea.Text

TextBox1.Text = TextBox1.Text & Textarea.Text & " XOR "

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

End If

End Sub

Private Sub Button20\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button20.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

value = value \* value

TextBox1.Text = Textarea.Text & "^2"

Else

value = 0

value = Textarea.Text

value = value \* value

TextBox1.Text = Textarea.Text & " ^2 "

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

Textarea.Text = value

End If

End Sub

Private Sub Button22\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button22.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

value = value \* value \* value

TextBox1.Text = Textarea.Text & "^3"

Else

value = 0

value = Textarea.Text

value = value \* value \* value

TextBox1.Text = Textarea.Text & " ^3"

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

Textarea.Text = value

End If

End Sub

Private Sub Button24\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button24.Click

'to be done

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

TextBox1.Text = Textarea.Text & " ^ "

Textarea.Clear()

Else

value = Math.Pow(value, Textarea.Text)

TextBox1.Text = TextBox1.Text & " ^ " & Textarea.Text

Textarea.Clear()

Textarea.Text = value

End If

'TextBox1.Text = value & "/"

End If

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

If Textarea.Text <> "" Then

If value = 0 Then

value = Textarea.Text

value = 1 / value

TextBox1.Text = "1/" & Textarea.Text

Else

value = 0

value = Textarea.Text

value = 1 / value

TextBox1.Text = "1/" & Textarea.Text

End If

'TextBox1.Text = value & "/"

Textarea.Clear()

Textarea.Text = value

End If

End Sub

Private Sub RadioDec\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioDec.CheckedChanged

btn2.Enabled = True

btn3.Enabled = True

btn4.Enabled = True

btn5.Enabled = True

btn6.Enabled = True

btn7.Enabled = True

btn8.Enabled = True

btn9.Enabled = True

'Me.BackColor = Color.AliceBlue

If Textarea.Text <> "" Then

Dim dec As Integer = 0, i As Integer = 0, p As Integer = 0, r As Integer = 0, t As Integer

t = 0

t = Textarea.Text

'Textarea.Text = Ns

Dim others As String

others = Textarea.Text

Textarea.Clear()

Dim ss As String = Ns

Console.WriteLine("Radio Dec: " & ss & "value is " & t)

If Ns = "Bin" Then

' bin to dec

While t > 0

r = t Mod 10

t = t / 10

p = Math.Pow(2, i)

dec = dec + (p \* r)

i += 1

Console.WriteLine("r " & r & "t " & t & "p " & p & " dec" & dec)

End While

Textarea.Text = dec

Console.WriteLine("dec value is : " & dec)

Ns = "Dec"

ElseIf Ns = "Hex" Then

'hex to dec

others = "&H" & others

Dim deca As Long

deca = Val(others)

Textarea.Text = dec

Console.WriteLine("dec value is : " & dec)

ElseIf Ns = "Oct" Then

'oct to dec

Dim deca As Long

others = "&O" & others

deca = Val(others)

Textarea.Text = dec

Console.WriteLine("dec value is : " & dec)

Else

Ns = "Dec"

End If

Else

Ns = "Dec"

End If

'Ns = "Dec"

End Sub

Private Sub RadioBin\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioBin.CheckedChanged

btn2.Enabled = False

btn3.Enabled = False

btn4.Enabled = False

btn5.Enabled = False

btn6.Enabled = False

btn7.Enabled = False

btn8.Enabled = False

btn9.Enabled = False

'btn2.Enabled = False

'Me.BackColor = Color.Brown

Dim bin As Integer = 0, i As Integer = 0, p As Integer = 0, r As Integer = 0, t As Integer

If Textarea.Text <> "" Then

t = 0

t = Textarea.Text

'Textarea.Text = Ns

Textarea.Clear()

Dim s As String = Ns

Console.WriteLine("Radio Bin: " & s)

If Ns = "Dec" Then

'dec to bin

While t > 0

r = t Mod 2

t = t \ 2

bin = bin \* 10 + r

End While

Textarea.Text = bin

Console.WriteLine("bin value is : " & bin)

Ns = "Bin"

ElseIf Ns = "Hex" Then

'hex to bin

ElseIf Ns = "Oct" Then

'oct to bin

Else

Ns = "Bin"

End If

Ns = "Bin"

End If

'Ns = "Bin"

End Sub

Private Sub RadioHex\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioHex.CheckedChanged

If btn2.Enabled = False Then

btn2.Enabled = True

btn3.Enabled = True

btn4.Enabled = True

btn5.Enabled = True

btn6.Enabled = True

btn7.Enabled = True

btn8.Enabled = True

btn9.Enabled = True

End If

'Me.BackColor = Color.AliceBlue

If Textarea.Text <> "" Then

Dim dec As Integer = 0, i As Integer = 0, p As Integer = 0, r As Integer = 0, t As Long

t = 0

t = Textarea.Text

Dim hexa As String

'Textarea.Text = Ns

Textarea.Clear()

Dim ss As String = Ns

Console.WriteLine("Radio HEx: " & ss & "value is " & t)

If Ns = "Bin" Then

' bin to hex

'While t > 0

'r = t Mod 10

' t = t / 10

' p = Math.Pow(2, i)

' dec = dec + (p \* r)

' i += 1

' Console.WriteLine("r " & r & "t " & t & "p " & p & " dec" & dec)

' End While

'Textarea.Text = dec

' Console.WriteLine("dec value is : " & dec)

' Ns = "Dec"

ElseIf Ns = "Dec" Then

'Dec to hex

hexa = Hex(t)

Textarea.Text = hexa

Console.WriteLine("hexa value is : " & hexa)

ElseIf Ns = "Oct" Then

'oct to hex

Else

Ns = "Hex"

End If

Ns = "Hex"

Else

Ns = "Hex"

End If

End Sub

Private Sub RadioOct\_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RadioOct.CheckedChanged

If btn2.Enabled = False Then

btn2.Enabled = True

btn3.Enabled = True

btn4.Enabled = True

btn5.Enabled = True

btn6.Enabled = True

btn7.Enabled = True

btn8.Enabled = True

btn9.Enabled = True

End If

'Me.BackColor = Color.AliceBlue

If Textarea.Text <> "" Then

Dim dec As Integer = 0, i As Integer = 0, p As Integer = 0, r As Integer = 0, t As Long

t = 0

t = Textarea.Text

Dim octa As String

'Textarea.Text = Ns

Textarea.Clear()

Dim ss As String = Ns

Console.WriteLine("Radio Oct: " & ss & "value is " & t)

If Ns = "Bin" Then

' bin to hex

'While t > 0

'r = t Mod 10

' t = t / 10

' p = Math.Pow(2, i)

' dec = dec + (p \* r)

' i += 1

' Console.WriteLine("r " & r & "t " & t & "p " & p & " dec" & dec)

' End While

'Textarea.Text = dec

' Console.WriteLine("dec value is : " & dec)

' Ns = "Dec"

ElseIf Ns = "Dec" Then

'Dec to Oct

octa = Oct(t)

Textarea.Text = octa

Console.WriteLine("octa value is : " & octa)

ElseIf Ns = "HEx" Then

'oct to hex

Else

Ns = "Oct"

End If

Ns = "Oct"

Else

Ns = "Oct"

End If

End Sub

Private Sub btnConv\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnConv.Click

Form2.Show()

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

Dim temp, temp1 As Integer

temp = Textarea.Text

temp1 = temp

temp = Math.Sqrt(temp)

TextBox1.Text = "Square root of " & temp1 & " is"

Textarea.Text = temp

End Sub

Private Sub btndot\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndot.Click

If Textarea.Text.Contains(".") Then

Else

Textarea.Text = Textarea.Text & "."

End If

End Sub

End Class

**Form2.vb**

Public Class Form2

Dim t As String, r As Integer = 0, bin As Integer

Dim octa, hexa As String

' Decimal to

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

'dec to bin

t = TextBox1.Text

Dim temp As Integer = t

While t > 0

r = t Mod 2

t = t \ 2

bin = bin \* 10 + r

End While

TextBox2.Text = temp & " in Binary is "

TextBox1.Text = bin

Console.WriteLine("bin value is : " & bin)

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

t = TextBox1.Text

octa = Oct(t)

TextBox2.Text = t & " in Octal is "

TextBox1.Text = octa

Console.WriteLine("octa value is : " & octa)

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

t = TextBox1.Text

hexa = Hex(t)

TextBox1.Text = hexa

TextBox2.Text = t & " in Hexadecimal is "

Console.WriteLine("hexa value is : " & hexa)

End Sub

Function Bin2Dec(ByVal bin As Integer) As Integer

Dim t As Integer, r As Integer = 0, dec As Integer = 0, i As Integer = 0, p As Integer

t = bin

While t > 0

r = t Mod 10

t = t / 10

p = Math.Pow(2, i)

dec = dec + (p \* r)

i += 1

'Console.WriteLine("r " & r & "t " & t & "p " & p & " dec" & dec)

End While

' Textarea.Text = dec

Console.WriteLine("dec value is : " & dec)

Return dec

End Function

Function Dec2Bin(ByVal dec As Integer) As Integer

'dec to bin

Dim t As Integer, r As Integer = 0, bin As Integer = 0

t = dec

While t > 0

r = t Mod 2

t = t \ 2

bin = bin \* 10 + r

End While

'Textarea.Text = bin

Console.WriteLine("bin value is : " & bin)

'Ns = "Bin"

Return bin

End Function

' Bin to

Private Sub Button6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click

Dim dec, bina As Integer

bina = TextBox1.Text

dec = Bin2Dec(bina)

TextBox1.Text = dec

TextBox2.Text = bina & " in decimal is "

Console.WriteLine("bin to dec is " & dec)

End Sub

Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

Dim bina, temp As Integer

bina = TextBox1.Text

temp = Bin2Dec(bina)

octa = Oct(temp)

TextBox1.Text = octa

TextBox2.Text = bina & " in Octal is "

End Sub

Private Sub Button5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click

Dim bina, temp As Integer

bina = TextBox1.Text

temp = Bin2Dec(bina)

hexa = Hex(temp)

TextBox1.Text = hexa

TextBox2.Text = bina & " in hexadecimal is "

End Sub

'Hex to

Private Sub Button9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button9.Click

hexa = "&H" & TextBox1.Text

Dim dec, bina As Long

dec = Val(hexa)

bina = Dec2Bin(dec)

TextBox1.Text = bina

TextBox2.Text = hexa & " in binary is "

End Sub

Private Sub Button8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button8.Click

hexa = "&H" & TextBox1.Text

Dim dec As Long

dec = Val(hexa)

TextBox1.Text = dec

TextBox2.Text = hexa & " in decimal is "

End Sub

Private Sub Button7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click

hexa = "&H" & TextBox1.Text

Dim dec As Long

dec = Val(hexa)

octa = Oct(dec)

TextBox1.Text = octa

TextBox2.Text = hexa & " in octal is "

End Sub

''Octal to

Private Sub Button12\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button12.Click

octa = "&O" & TextBox1.Text

Dim dec, bina As Long

dec = Val(octa)

bina = Dec2Bin(dec)

TextBox1.Text = bina

TextBox2.Text = octa & " in binary is "

End Sub

Private Sub Button10\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button10.Click

octa = "&O" & TextBox1.Text

Dim dec As Long

dec = Val(octa)

TextBox1.Text = dec

TextBox2.Text = octa & " in Decimal is "

End Sub

Private Sub Button11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button11.Click

octa = "&O" & TextBox1.Text

Dim dec As Long

dec = Val(octa)

hexa = Hex(dec)

TextBox1.Text = hexa

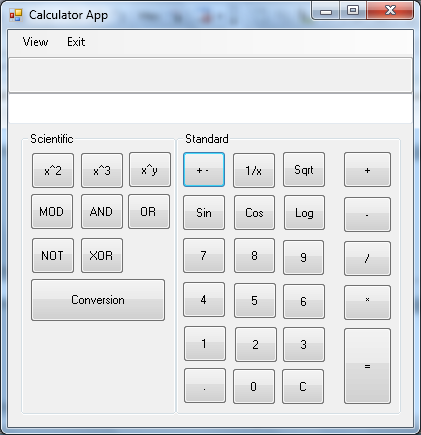
TextBox2.Text = octa & " in Hexadecimal is "

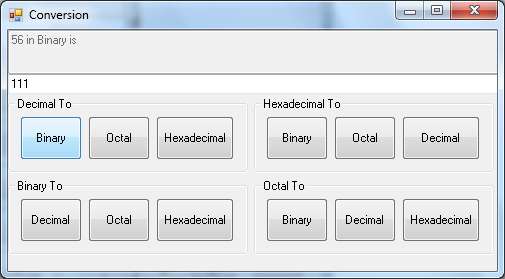
End Sub

Private Sub Form2\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

End Sub

End Class





## Notepad

**Soln : Form1.vb**

Public Class Form1

Public count As Integer

Private Sub NewToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripMenuItem.Click

'Check if there's text added to the textbox

If RichTextBox1.Modified Then

'If the text of notepad changed, the program will ask the user if they want to save the changes

Dim ask As MsgBoxResult

ask = MsgBox("Do you want to save the changes", MsgBoxStyle.YesNoCancel, "New Document")

If ask = MsgBoxResult.No Then

RichTextBox1.Clear()

ElseIf ask = MsgBoxResult.Cancel Then

ElseIf ask = MsgBoxResult.Yes Then

SaveFileDialog1.ShowDialog()

My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, RichTextBox1.Text, False)

RichTextBox1.Clear()

End If

'If textbox's text is still the same, notepad will open a new page:

Else

RichTextBox1.Clear()

End If

End Sub

Private Sub OpenToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripMenuItem.Click

'Check if there's text added to the textbox

If RichTextBox1.Modified Then

'If the text of notepad changed, the program will ask the user if they want to save the changes

Dim ask As MsgBoxResult

ask = MsgBox("Do you want to save the changes", MsgBoxStyle.YesNoCancel, "Open Document")

If ask = MsgBoxResult.No Then

OpenFileDialog1.ShowDialog()

RichTextBox1.Text = My.Computer.FileSystem.ReadAllText(OpenFileDialog1.FileName)

ElseIf ask = MsgBoxResult.Cancel Then

ElseIf ask = MsgBoxResult.Yes Then

SaveFileDialog1.ShowDialog()

My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, RichTextBox1.Text, False)

RichTextBox1.Clear()

End If

Else

'If textbox's text is still the same, notepad will show the OpenFileDialog

OpenFileDialog1.ShowDialog()

Try

RichTextBox1.Text = My.Computer.FileSystem.ReadAllText(OpenFileDialog1.FileName)

Catch ex As Exception

End Try

End If

End Sub

Private Sub SaveToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripMenuItem.Click

'Dim dlg As SaveFileDialog = New SaveFileDialog

''dlg.Title = "Save"

'dlg.DefaultExt = ".txt"

'dlg.FileName = "\*"

'dlg.AddExtension = True

'dlg.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*"

'dlg.ShowDialog()

'' the application will check if the file already exists, if exists, it will ask the user if they want to replace it

'If My.Computer.FileSystem.FileExists(dlg.FileName) Then

' Dim ask As MsgBoxResult

' ask = MsgBox("File already exists, would you like to replace it?", MsgBoxStyle.YesNo, "File Exists")

' 'if the user decides not to replace the existing file

' If ask = MsgBoxResult.No Then

' dlg.ShowDialog()

' 'if the user decides to replace the existing file

' ElseIf ask = MsgBoxResult.Yes Then

' My.Computer.FileSystem.WriteAllText(dlg.FileName, RichTextBox1.Text, True)

' End If

' 'if the file doesn't exist

'Else

' Try

' RichTextBox1.SaveFile(SaveFileDialog1.FileName, RichTextBoxStreamType.RichText)

' 'My.Computer.FileSystem.WriteAllText(SaveFileDialog1.FileName, RichTextBox1.Text, True)

' Catch ex As Exception

' End Try

'End If

Dim saveMe As New SaveFileDialog

saveMe.Title = "Save"

saveMe.DefaultExt = ".txt"

saveMe.FileName = "\*"

saveMe.AddExtension = True

saveMe.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*"

If saveMe.ShowDialog() = Windows.Forms.DialogResult.OK Then

RichTextBox1.SaveFile(saveMe.FileName,RichTextBoxStreamType.PlainText)

Else

MsgBox("File Not save")

End If

End Sub

Private Sub UndoToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles UndoToolStripMenuItem.Click

If RichTextBox1.CanUndo Then

RichTextBox1.Undo()

Else

End If

End Sub

Private Sub CutToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CutToolStripMenuItem.Click

My.Computer.Clipboard.Clear()

If RichTextBox1.SelectionLength > 0 Then

My.Computer.Clipboard.SetText(RichTextBox1.SelectedText)

End If

RichTextBox1.SelectedText = ""

End Sub

Private Sub CopyToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CopyToolStripMenuItem.Click

My.Computer.Clipboard.Clear()

If RichTextBox1.SelectionLength > 0 Then

My.Computer.Clipboard.SetText(RichTextBox1.SelectedText)

End If

End Sub

Private Sub PasteToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PasteToolStripMenuItem.Click

If RichTextBox1.SelectionLength > 0 Then

If My.Computer.Clipboard.ContainsText Then

RichTextBox1.Paste()

End If

End If

End Sub

Private Sub FindToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FindToolStripMenuItem.Click

Dim a As String

Dim b As String

a = InputBox("Enter text to be found")

b = InStr(RichTextBox1.Text, a)

If b Then

RichTextBox1.Focus()

RichTextBox1.SelectionStart = b - 1

RichTextBox1.SelectionLength = Len(a)

Else

MsgBox("Text not found.")

End If

End Sub

Private Sub FontToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FontToolStripMenuItem.Click

FontDialog1.ShowDialog()

RichTextBox1.Font = FontDialog1.Font

End Sub

Private Sub FontColourToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles FontColourToolStripMenuItem.Click

ColorDialog1.ShowDialog()

RichTextBox1.ForeColor = ColorDialog1.Color

End Sub

Private Sub BackColourToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BackColourToolStripMenuItem.Click

ColorDialog1.ShowDialog()

RichTextBox1.BackColor = ColorDialog1.Color

End Sub

Private Sub PrintToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PrintToolStripMenuItem.Click

PrintDialog1.ShowDialog()

End Sub

Private Sub ExitToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ExitToolStripMenuItem.Click

Dim response As MsgBoxResult

response = MsgBox("Do you want to close the applications?", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "Confirm")

If response = MsgBoxResult.Yes Then

Me.Dispose()

ElseIf response = MsgBoxResult.No Then

Exit Sub

End If

End Sub

Private Sub SaveAsToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveAsToolStripMenuItem.Click

Try

'Dim dlg As SaveFileDialog = New SaveFileDialog

' dlg.Title = "Save As"

'dlg.Filter = "TextDocuments(\*.txt)"

'If dlg.ShowDialog() = System.Windows.Forms.DialogResult.OK Then

'RichTextBox1.SaveFile(dlg.FileName, RichTextBoxStreamType.RichText)

'End If

Dim saveMe As New SaveFileDialog

saveMe.Title = "SaveAs"

saveMe.DefaultExt = ".txt"

saveMe.FileName = "\*"

saveMe.AddExtension = True

saveMe.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*"

If saveMe.ShowDialog() = Windows.Forms.DialogResult.OK Then

RichTextBox1.SaveFile(saveMe.FileName, RichTextBoxStreamType.PlainText)

Else

MsgBox("File Not save")

End If

Catch ex As Exception : End Try

End Sub

Private Sub LeftToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LeftToolStripMenuItem.Click

RichTextBox1.SelectionAlignment = HorizontalAlignment.Left

End Sub

Private Sub CenterToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CenterToolStripMenuItem.Click

RichTextBox1.SelectionAlignment = HorizontalAlignment.Center

End Sub

Private Sub RightToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RightToolStripMenuItem.Click

RichTextBox1.SelectionAlignment = HorizontalAlignment.Right

End Sub

Private Sub SelectAllToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SelectAllToolStripMenuItem.Click

RichTextBox1.SelectAll()

End Sub

Private Sub RichTextBox1\_SelectionChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles RichTextBox1.SelectionChanged

If RichTextBox1.SelectedText <> "" Then

CopyToolStripMenuItem.Enabled = True

CutToolStripMenuItem.Enabled = True

SelectAllToolStripMenuItem.Enabled = True

DeleteToolStripMenuItem.Enabled = True

ElseIf RichTextBox1.SelectedText = "" Then

CopyToolStripMenuItem.Enabled = False

CutToolStripMenuItem.Enabled = False

' SelectAllToolStripMenuItem.Enabled = False

DeleteToolStripMenuItem.Enabled = False

End If

End Sub

Private Sub Form1\_FormClosed(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed

Dim response As MsgBoxResult

If RichTextBox1.Text <> "" Then

MsgBox("Save your File", MsgBoxStyle.YesNo)

If response = MsgBoxResult.Yes Then

Dim saveMe As New SaveFileDialog

saveMe.Title = "Save"

saveMe.DefaultExt = ".txt"

saveMe.FileName = "\*"

saveMe.AddExtension = True

saveMe.Filter = "Text files(\*.txt)|\*.txt|All files(\*.\*)|\*.\*"

If saveMe.ShowDialog() = Windows.Forms.DialogResult.OK Then

RichTextBox1.SaveFile(saveMe.FileName, RichTextBoxStreamType.PlainText)

Else

MsgBox("File Not save")

End If

End If

End If

End Sub

Private Sub NotepadToolStripMenuItem\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NotepadToolStripMenuItem.Click

About.Show()

End Sub

End Class

