Contents

[Console Applications 3](#_Toc461626914)

[1. Program to Find Factorial of a number 3](#_Toc461626915)

[2. Program to Find Sum of Fibonacci series upto n 4](#_Toc461626916)

[3. program to find the reverse of a number 4](#_Toc461626917)

[4. program to find the sum of the digits of a number 4](#_Toc461626918)

[5. program to check whether the number is a prime number 4](#_Toc461626919)

[6. program to check whether the number it is an armstrong number 5](#_Toc461626920)

[7. program to find all the factors of a number 5](#_Toc461626921)

[8. program to find a series of Armstrong numbers from 1 to ‘n ’ 5](#_Toc461626922)

[9. program to find a series of prime numbers from 2 to ‘n’ 5](#_Toc461626923)

[10. program to find whether a number ‘n’ is an Automorphic number or not 5](#_Toc461626924)

[ARRAY CONSOLE APPLICATIONS 6](#_Toc461626925)

[1. Program to Display sum and average of all elements in an array. 6](#_Toc461626926)

[2. Program to Display the values in the array, along with the largest and smallest element in the array. 6](#_Toc461626927)

[3. Program to perform searching for an element in an array. 6](#_Toc461626928)

[4. Program to store numeric values in an ‘m x n’ matrix and display the matrix and the sum of its diagonal elements. 6](#_Toc461626929)

[5. Program to store numeric values in two ‘m x n’ matrix and display the two matrices along with their sum and difference 6](#_Toc461626930)

[ARRAY GUI APPLICATIONS 6](#_Toc461626931)

[1. Program to Display the values in the array, the sum and average of all the elements in the array. 6](#_Toc461626932)

[2. Program to 6](#_Toc461626933)

[3. Program to 6](#_Toc461626934)

[4. Program to 6](#_Toc461626935)

[5. Program to 6](#_Toc461626936)

# 

# 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

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()

End While

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

Console.Read()

End Sub

End Module

## 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

## 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

## 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

## 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

## 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

## 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

## 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

## 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

# 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

## 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)

## 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)

## 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)

## 5. Program to store numeric values in two ‘m x n’ matrix. 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. 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

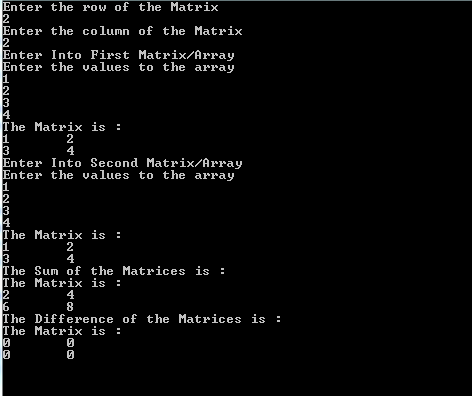
Next

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.

## 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.

## 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.

## 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.

# 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**

**‘Input Form**

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

**‘Dynamic Form**

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

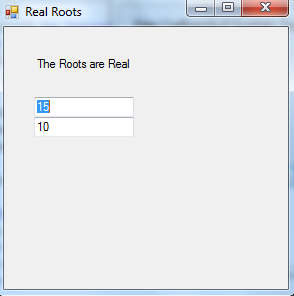
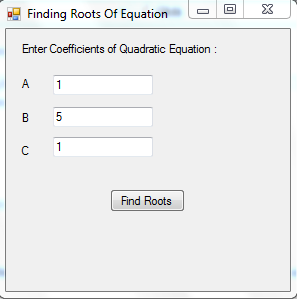
End With

Me.Controls.Add(label)

Me.Text = "Imaginary Roots"

End If

End Sub

****End Class

**OUTPUT**