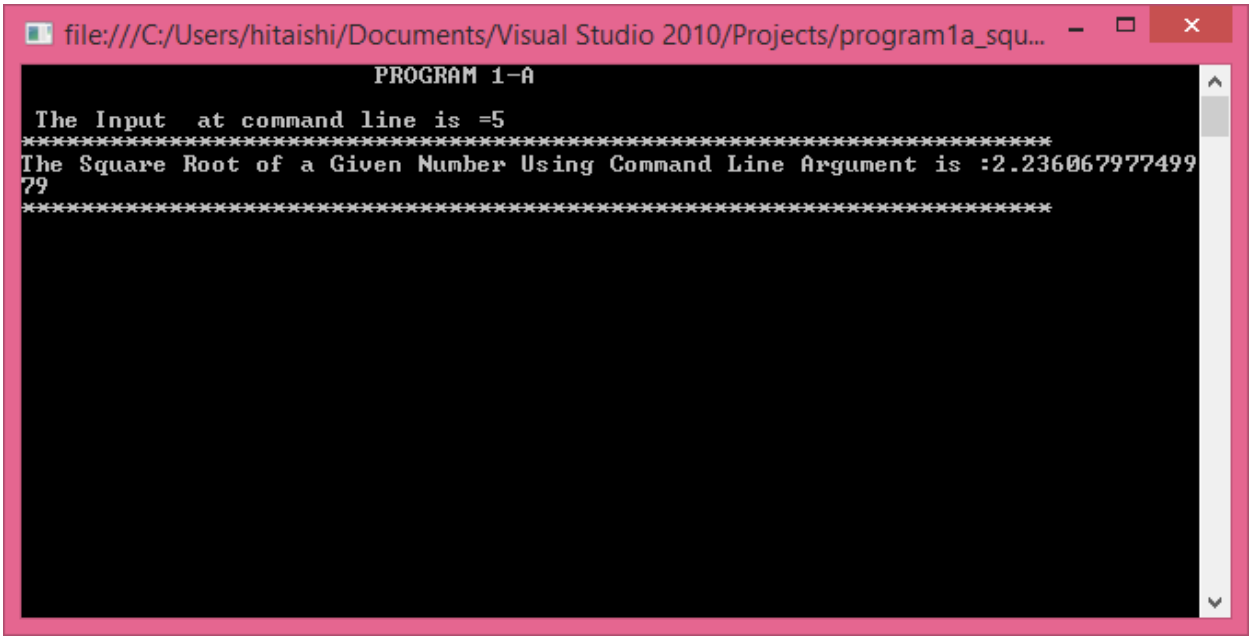
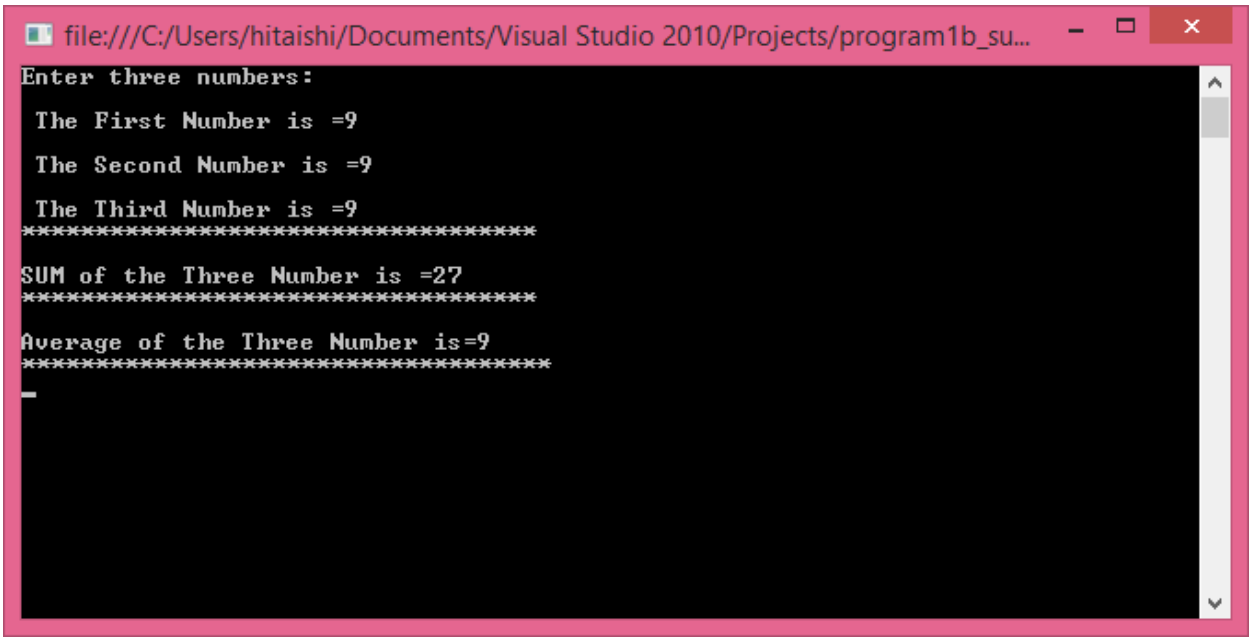
Program 1. Write a Program in C# to demonstrate Command line arguments processing.

```
// A Program to find the square root of a given number in C# to demonstrate command line
arguments
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace sq
{
   class program1_square
   {
      public static void Main(string[] args)
      {
         //Declare variable
         double argsValue = 0;
         double sqrtValue = 0;
         //check the length of command line argument
         if (args.Length == 0)
         {
            Console.WriteLine("****************************************");
            Console.WriteLine("There is no command line Argument defined");
```

```
            Console.WriteLine("*****************************************");
            Console.ReadLine();
            return;
        }
        //Find square root of the number
          using math argsValue =
          double.Parse(args[0].ToString());
          sqrtValue = Math.Sqrt(argsValue);
        //Display sqrt of the number
        Console.WriteLine("\t
        \t\tPROGRAM 1-A");
        Console.WriteLine("\n The Input at command line is =" + argsValue);
        Console.WriteLine("*****************************************************
        "); Console.WriteLine("The Square Root of a Given Number Using Command Line
                                Argument is :{0}", sqrtValue);
        Console.WriteLine("*****************************************************"
        ); Console.ReadLine();

    }
  }

}
```






```
using System;
namespace labproject
{
   class program1
   {
      /* c# program -Command line arguements */

                  public static void Main(String[] args)
                  {
                        Console.WriteLine ("The Number of arguments are : {0}",

                  args.Length);
                        for (int i = 0; i < args.Length; i++)
                        Console.WriteLine ("arguement {0} is {1}",i+1,args[i]);
         Console.ReadLine();
                  }
   }
}


using System;

namespace MyApplication
{
 class Program
  {
```

```
    static void Main(string[] args)
    {
      int x = 5;
      int y = 6;
      int sum = x + y;
      Console.WriteLine(sum); // Print the sum of x + y
    }
  }
}
```

Program 2. Find the sum of all the elements present in a jagged array of 3 inner arrays.

```
using System;
namespace labproject
{
    class program4
    {
        /* c# program -Jagged Array */

                public static void Main ()
                {
                        int[][] jag;
                        int i, j, var, sum = 0;
                        Console.WriteLine ("Enter the number of rows");
                        int row = int.Parse (Console.ReadLine ());
                        jag = new int[row][];
                        for (i = 0; i < row; i++) {
                                Console.WriteLine ("Enter the number of elements in row {0} :", i
+ 1);
                                var = int.Parse (Console.ReadLine ());
                                jag [i] = new int[var];
                                Console.WriteLine (" Enter the {0} Values ",var);
                                for (j = 0; j < var; j++) {
                                        jag [i] [j] = int.Parse (Console.ReadLine ());
                                }
                                Console.WriteLine ();
                        }
                        Console.WriteLine ("jag[{0}][]:",row);
                        for (i = 0; i < row; i++) {
                                for (j = 0; j < jag [i].Length; j++) {
                                        Console.Write (" " + jag [i] [j]);
                                        sum = sum + jag [i] [j];
                                }
                        }
                        Console.WriteLine (sum);
        Console.ReadLine();
```

```
        }
    }
}
```

Program 3: Write a Program in C# to demonstrate Array Out of Bound Exception using Try, Catch and Finally blocks.

```
using System;
namespace labproject
{
   class program5
   {
      /* c# program -try {} catch{} finally */
            public static void Main()
            {
                  Console.WriteLine ("Enter the numerator");
                  int p = int.Parse (Console.ReadLine ());
                  Console.WriteLine ("Enter the Denominator");
                  int q = int.Parse (Console.ReadLine ());
                  try
                  {
                        int r=p/q;
                        Console.WriteLine("The Value of Quotient is {0}",r);
                  }
                  catch(Exception e) {
                        Console.WriteLine (e.Message);
                  }
                  finally{
                        Console.WriteLine ("Execution Completed");
                  }
         Console.ReadLine();

            }
      }
}
```

```
using System;

class ExceptionDemo
{
   static void Main(string[] args)
   {
      int[] intArray = new int[5] { 50,40,30,20,10 };
      int iLoop   = 0;
```

```
        int sum     = 0;

        try
        {
           for (iLoop = 0; iLoop <= 5; iLoop++)
           {
              sum += intArray[iLoop];
           }
           Console.WriteLine("Sum of array elements:" + sum);
        }
        catch (IndexOutOfRangeException e)
        {
           Console.WriteLine(e.Message);
        }
     }
}
```

Program 4: Write a Program to Demonstrate Use of Virtual and override key words in C# with a simple program

```
using System;
namespace labproject
{
        /* c# program -Vertual and Override*/
           class BaseClass
           {
                    public virtual string city()
                    {
                             return "BMSIT";
                    }
           }

           class DerivedClass : BaseClass
           {
                    public override string city()
                    {
                             return "BMSCE";
                    }
           }
           class Program6
           {
                    public static void Main()
                    {
                             DerivedClass o = new DerivedClass ();
                             string city = o.city ();
                             Console.WriteLine (" College Name is : {0}", city);
           Console.ReadLine();
                    }
```

```
        }
    }
```

Program 5: Write a Program in C# to create and implement a Delegate for any two arithmetic operations.

```
using System;
namespace labproject
{
public abstract class program
{
public abstract int mul(int a, int b);
}
public class demo : program
{
public override int mul(int a, int b)
{
return a * b;
}
}
public class main
{
public static void Main()
{
demo d = new demo(); int j = d.mul(20, 30);
Console.WriteLine("\n**************************************"); Console.WriteLine("The Multiplication of {0} * {1} = {2}", 20, 30, j);
Console.WriteLine("\n**************************************"); Console.ReadLine();
}
}
}
```

Program 6: Write a Program in C# to demonstrate abstract class and abstract methods in C#.

```
using System;
public abstract class Shape
{
   public abstract void draw();
}
public class Rectangle : Shape
{
   public override void draw()
   {
      Console.WriteLine("drawing rectangle...");
```

```
    }
}
public class Circle : Shape
{
    public override void draw()
    {
        Console.WriteLine("drawing circle...");
    }
}
public class TestAbstract
{
    public static void Main()
    {
        Shape s;
        s = new Rectangle();
        s.draw();
        s = new Circle();
        s.draw();
    }
}
```