1. Program to find the sum of digits of a number

```
using System;

class Program
{
   static void Main()
   {
      int num, sum = 0;
      Console.Write("Enter a number: ");
      num = int.Parse(Console.ReadLine());

      while (num != 0)
      {
         sum += num % 10;
         num /= 10;
      }

      Console.WriteLine("Sum of digits = " + sum);
   }
}
```

C# Program to Print Binary Equivalent of an Integer using Recursion

```
using System;

class Program
{
   static void Main()
   {
      Console.Write("Enter a number: ");
      int number = int.Parse(Console.ReadLine());
```

```
        Console.Write("Binary equivalent: ");
        PrintBinary(number);
    }

    static void PrintBinary(int num)
    {
        if (num == 0)
            return;

        PrintBinary(num / 2);        // Recursive call with num divided by 2
        Console.Write(num % 2);      // Print remainder (0 or 1)
    }
}
```

Program to swap two numbers with and without a third number

With:

```
using System;

namespace Swap
{
    class Program
    {
        static void Main(string[] args)
        {
            int num1, num2, temp;

            Console.WriteLine("Name: Srushti Rajaram Shetake, Roll no: 252, Div: B");

            // Input from the user
            Console.Write("Enter number1: ");
```

```
            num1 = Convert.ToInt32(Console.ReadLine());

            Console.Write("Enter number2: ");
            num2 = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Before swapping: number1 = " + num1 + " and number2 = " + num2);

            // Swapping logic
            temp = num1;
            num1 = num2;
            num2 = temp;

            Console.WriteLine("After swapping: number1 = " + num1 + " and number2 = " + num2);

            Console.ReadKey();
        }
    }
}
```

Without:

```
using System;

namespace Swap
{
    class Program
    {
        static void Main(string[] args)
        {
            int num1, num2;
```

```
        Console.WriteLine("Name: Srushti Rajaram Shetake, Roll no: 252,
    Div: B");


        // Get input from the user
        Console.Write("Enter number1: ");
        num1 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Enter number2: ");
        num2 = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Before swapping: number1 = " + num1 + " and
    number2 = " + num2);

        // Swapping without using a third variable
        num1 = num1 + num2;
        num2 = num1 - num2;
        num1 = num1 - num2;

        Console.WriteLine("After swapping: number1 = " + num1 + " and
    number2 = " + num2);

        Console.ReadKey();
      }
   }
```

}**2.Calculate the factorial of a given number**

```
using System;
public class Factorial
```

```
{
   public static void Main(string[] args)
   {
      Console.WriteLine("Name: Srushti Rajaram Shetake, Roll no: 252, Div: B");
      int i, fact = 1, number;
      Console.Write("Enter any Number: ");
      number = int.Parse(Console.ReadLine());
      for (i = 1; i <= number; i++)
      {
         fact = fact * i;
      }
      Console.Write("Factorial of " + number + " is: " + fact);
   }
}
```

Print odd numbers in a given range

```
using System;

class Program
{
   static void Main()
   {
      Console.Write("Enter starting number: ");
      int start = int.Parse(Console.ReadLine());

      Console.Write("Enter ending number: ");
      int end = int.Parse(Console.ReadLine());

      Console.WriteLine("Odd numbers between {0} and {1} are:", start, end);

      for (int i = start; i <= end; i++)
```

```
        {
            if (i % 2 != 0)  // Check if number is odd
            {
                Console.Write(i + " ");
            }
        }
    }
}
```

Print all multiples of 17 which are less than 200

```
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("Multiples of 17 less than 200:");

        for (int i = 17; i < 200; i += 17)
        {
            Console.Write(i + " ");
        }
    }
}
```

3.Print the Fibonacci series up to the first 6 numbers

```
        using System;

        class Program
        {
            static void Main(string[] args)
```

```
{
    // 1. Print the Fibonacci series up to the first 6 numbers
    Console.WriteLine("Fibonacci Series (First 6 Numbers):");
    int n1 = 0, n2 = 1, n3;
    Console.Write(n1 + " " + n2 + " ");
    for (int i = 2; i < 6; i++)
    {
        n3 = n1 + n2;
        Console.Write(n3 + " ");
        n1 = n2;
        n2 = n3;
    }

    Console.WriteLine("\n");

    // 2. Sum of digits of a number
    Console.Write("Enter a number to find sum of its digits: ");
    int num = Convert.ToInt32(Console.ReadLine());
    int sum = 0, temp = num;

    while (temp > 0)
    {
        sum += temp % 10;
        temp /= 10;
    }

    Console.WriteLine("Sum of digits: " + sum);

    // 3. Calculate the factorial of a number
    Console.Write("\nEnter a number to find its factorial: ");
    int factNum = Convert.ToInt32(Console.ReadLine());
```

```
            long factorial = 1;

            for (int i = 1; i <= factNum; i++)
            {
               factorial *= i;
            }

            Console.WriteLine("Factorial of " + factNum + " is " + factorial);
         }
      }
```

5.C# Program to overload unary "-" Operator

```
using System;
using System.Runtime.Remoting.Messaging;

namespace Calculator
{
   internal class Calculator
   {
      public int number1, number2;
      public Calculator(int num1, int num2)
      {
         number1 = num1;
         number2 = num2;
      }
      public static Calculator operator -(Calculator c1)
      {
         c1.number1 = -c1.number1;
         c1.number2 = -c1.number2;
         return c1;
```

```
        }
        public void print()
        {
           Console.WriteLine("number1= " + number1);
           Console.WriteLine("number2= " + number2);

        }
        class program
        {


           static void Main(string[] args)
           {
              Console.WriteLine("SRUSHTI SHETAKE roll_no252");
              Calculator calc = new Calculator(-10, -25);
              calc = -calc;
              calc.print();
           }
        }
    }
}
```

4.**C# Program to Overload Binary Plus "+" Operator**

```
using System;



namespace overload
{ class BinaryOverloading
    {
        public int number1=0;
        public BinaryOverloading() { }
```

```
public BinaryOverloading(int num1)
{
   number1 = num1;

}

public static BinaryOverloading operator +(BinaryOverloading c1, BinaryOverloading c2)
{
   BinaryOverloading c3 = new BinaryOverloading(0);
   c3.number1 = c1.number1 + c2.number1;
   return c3;
}
public void print()
{
   Console.WriteLine(number1);



}
class program
{


   static void Main(string[] args)
   {
      Console.WriteLine("srushti shetake roll_no=252");
      BinaryOverloading c1 = new BinaryOverloading(250);
      BinaryOverloading c2 = new BinaryOverloading(50);
      BinaryOverloading c3 = new BinaryOverloading();
      c3 = c1 + c2;
      c1.print();
      c2.print();
```

```
            c3.print();}}}}
```

6.C# Program to overload ++ operator.

```
using System;

class Counter
{
   public int Value;

   public Counter(int value)
   {
      Value = value;
   }

   // Overloading the ++ operator
   public static Counter operator ++(Counter c)
   {
      c.Value += 1;
      return c;
   }

   public void Display()
   {
      Console.WriteLine("Value: " + Value);
   }
}
class Program
{
   static void Main()
   {
```

```
        Counter c1 = new Counter(5);

        Console.WriteLine("Before increment:");
        c1.Display();

        c1++;  // Calls overloaded ++ operator

        Console.WriteLine("After increment:");
        c1.Display();
    }
}
```

7.C# Program to find all substrings in a string

```
using System;

class Program
{
    static void Main()
    {
        Console.Write("Enter a string: ");
        string input = Console.ReadLine();

        Console.WriteLine("All substrings are:");

        for (int i = 0; i < input.Length; i++)
        {
            for (int j = 1; j <= input.Length - i; j++)
```

```
        {
            string substring = input.Substring(i, j);
            Console.WriteLine(substring);
        }
    }
  }
}
```

C# Program to convert characters of a string to the opposite case

```
using System;

class Program
{
  static void Main()
  {
    Console.Write("Enter a string: ");
    string input = Console.ReadLine();
    string result = "";

    foreach (char c in input)
    {
      if (char.IsUpper(c))
        result += char.ToLower(c);
      else if (char.IsLower(c))
        result += char.ToUpper(c);
      else
        result += c; // Keep non-letter characters as they are
    }

    Console.WriteLine("Converted string: " + result);
  }
```

```
}
```

8.C# Program for searching a number in a given array.
C# Program to find all substrings in a string

```
using System;

class Program
{
   static void Main(string[] args)
   {
      // 1. Searching a number in a given array
      Console.WriteLine("Enter elements of the array (comma separated): ");
      string[] input = Console.ReadLine().Split(',');
      int[] arr = Array.ConvertAll(input, int.Parse);

      Console.Write("Enter number to search: ");
      int key = Convert.ToInt32(Console.ReadLine());

      bool found = false;
      for (int i = 0; i < arr.Length; i++)
      {
         if (arr[i] == key)
         {
            Console.WriteLine("Number found at index: " + i);
            found = true;
            break;
         }
      }

      if (!found)
         Console.WriteLine("Number not found in the array.");
```

```
        Console.WriteLine("\n-----------------------------------\n");

        // 2. Find all substrings in a string
        Console.Write("Enter a string to find all substrings: ");
        string str = Console.ReadLine();

        Console.WriteLine("All Substrings:");
        for (int i = 0; i < str.Length; i++)
        {
            for (int j = 1; j <= str.Length - i; j++)
            {
                Console.WriteLine(str.Substring(i, j));
            }
        }
    }
}
```

9.C# Program to demonstrate interface inheritance (The shape and colour program practiced in the practicale sesison)

```
using System;

interface IShape
{
    void Draw();
}

interface IColour
{
    void FillColour();
}
```

```
// Inheriting both interfaces
class Circle : IShape, IColour
{
    public void Draw()
    {
        Console.WriteLine("Drawing a Circle.");
    }

    public void FillColour()
    {
        Console.WriteLine("Filling Circle with Red Colour.");
    }
}

class Rectangle : IShape, IColour
{
    public void Draw()
    {
        Console.WriteLine("Drawing a Rectangle.");
    }

    public void FillColour()
    {
        Console.WriteLine("Filling Rectangle with Blue Colour.");
    }
}

class Program
{
    static void Main(string[] args)
```

```
    {
        IShape shape1 = new Circle();
        IColour colour1 = new Circle();

        IShape shape2 = new Rectangle();
        IColour colour2 = new Rectangle();

        Console.WriteLine("Circle:");
        shape1.Draw();
        colour1.FillColour();

        Console.WriteLine("\nRectangle:");
        shape2.Draw();
        colour2.FillColour();
    }
}
```

10. C# Program to implement Minimum 5 StringBuilder methods

```
using System;
using System.Text;

class Program
{
    static void Main()
    {
        StringBuilder sb = new StringBuilder("Hello");

        // 1. Append
        sb.Append(" World");
```

```
        Console.WriteLine("After Append: " + sb);

        // 2. Insert
        sb.Insert(5, ",");
        Console.WriteLine("After Insert: " + sb);

        // 3. Replace
        sb.Replace("World", "C#");
        Console.WriteLine("After Replace: " + sb);

        // 4. Remove
        sb.Remove(5, 1); // Removes the comma
        Console.WriteLine("After Remove: " + sb);

        // 5. Clear
        sb.Clear();
        Console.WriteLine("After Clear: \"" + sb + "\" (empty string)");
    }
}
```

12. Program that Implements Multidimensional & Jagged array.

```
using System;

class Program
{
    static void Main()
    {
Console.WriteLine("Name:Srushti Shetake,Roll no:252");
        // Multidimensional Array (2D)
        int[,] multiArray = {
            { 1, 2, 3 },
            { 4, 5, 6 },
```

```
        { 7, 8, 9 }
    };

    Console.WriteLine("Multidimensional Array:");
    for (int i = 0; i < multiArray.GetLength(0); i++)
    {
        for (int j = 0; j < multiArray.GetLength(1); j++)
        {
            Console.Write(multiArray[i, j] + " ");
        }
        Console.WriteLine();
    }

    Console.WriteLine();

    // Jagged Array (Array of Arrays)
    int[][] jaggedArray = new int[3][];
    jaggedArray[0] = new int[] { 1, 2 };
    jaggedArray[1] = new int[] { 3, 4, 5 };
    jaggedArray[2] = new int[] { 6, 7, 8, 9 };

    Console.WriteLine("Jagged Array:");
    for (int i = 0; i < jaggedArray.Length; i++)
    {
        for (int j = 0; j < jaggedArray[i].Length; j++)
        {
            Console.Write(jaggedArray[i][j] + " ");
        }
        Console.WriteLine();
    }
}
```

}