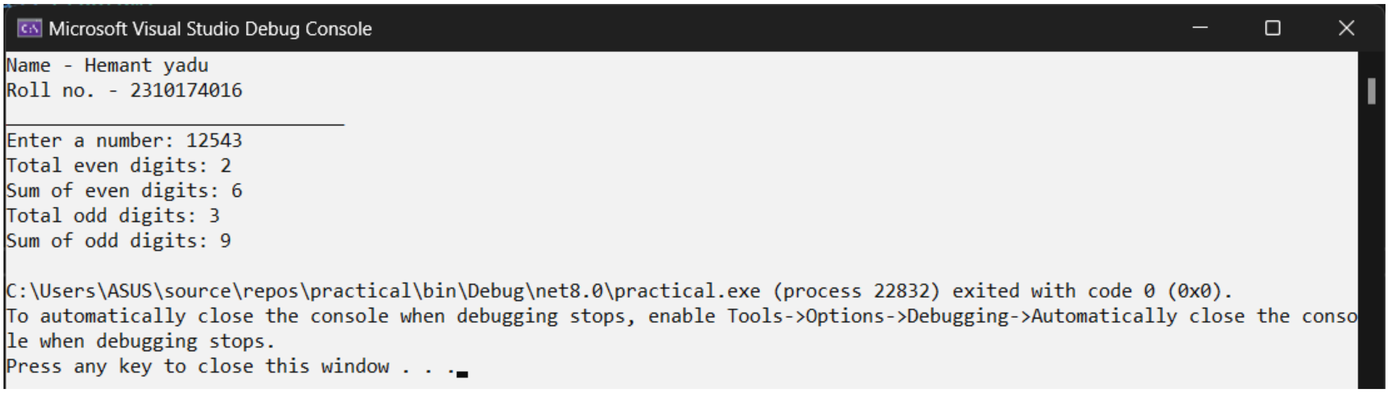
Practical No. 1

Write a C# program to find out the total number of an odd digits and even digits within the given number and print the sum of all odd digits and sum of even digits.

Code:

```
using System;
class Program1{
  static void Main(){
Console.Write("Name - Hemant yadu");
Console.Write("Roll no. - 2310174016");
Console.Write("______________________________");
    Console.Write("Enter a number: ");
    string input = Console.ReadLine();

    int oddCount = 0, evenCount = 0;
    int oddSum = 0, evenSum = 0;

    foreach (char digitChar in input){
      int digit = digitChar - '0';                                  // Convert character to integer

      if (digit % 2 == 0) // Check if even{
        evenCount++;
        evenSum += digit;
      }
      else{                                              // Odd
        oddCount++;
        oddSum += digit;
      }
    }
    Console.WriteLine($"Total even digits: {evenCount}");
    Console.WriteLine($"Sum of even digits: {evenSum}");
    Console.WriteLine($"Total odd digits: {oddCount}");
    Console.WriteLine($"Sum of odd digits: {oddSum}");
  }
}
```

Output:



```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
Enter a number: 12543
Total even digits: 2
Sum of even digits: 6
Total odd digits: 3
Sum of odd digits: 9

C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe (process 22832) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

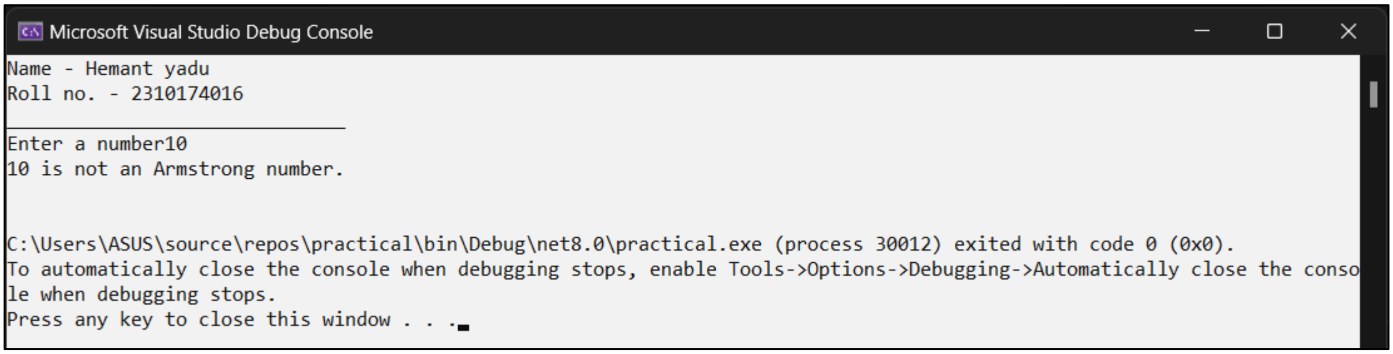
Practical No. 2

Write a C# program to input a number and check whether the number is Armstrong or not. (A number is Armstrong number if sum of each digit raised to(total digit is = number.

Code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace practical{
   internal class program2{
      public program2(){
         Console.Write("Enter a number");
         int number = int.Parse(Console.ReadLine());
         int sum = 0;
         int temp = number;
         int digits = number.ToString().Length;
         while (temp > 0){
            int digit = temp % 10;
            sum += (int)Math.Pow(digit, digits);
            temp /= 10;
         }
         if (sum == number){
            Console.WriteLine($"{number} is an Armstrong number.");
         }
         else
{
            Console.WriteLine($"{number} is not an Armstrong number. ");
         }
         Console.ReadLine();
      }}}
```

Output:



```
Name - Hemant yadu
Roll no. - 2310174016
________________________________
Enter a number10
10 is not an Armstrong number.


C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe (process 30012) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

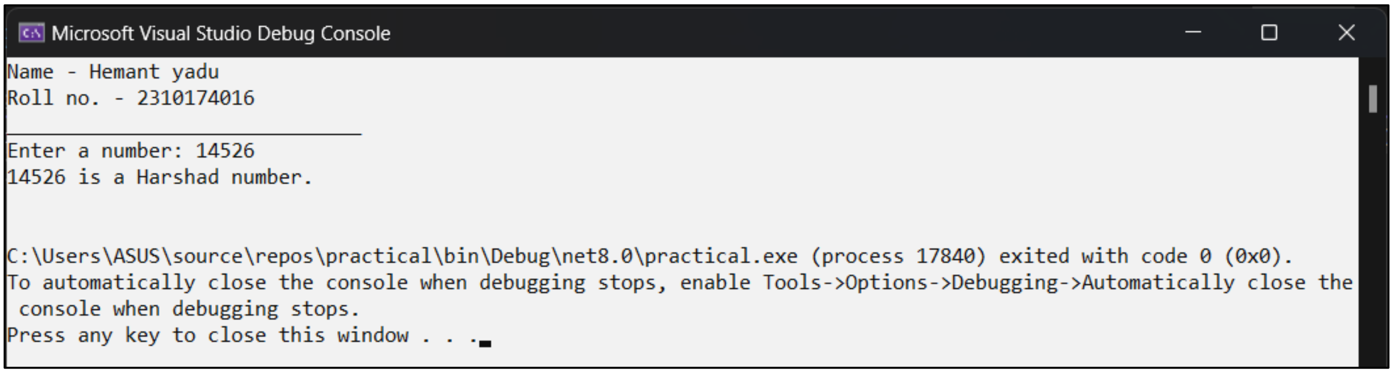
Practical No. 3

Write a C# program to check whether number is Harshad Number or not.

Code:

```
using System;
namespace practical{
   internal class prog3 {
      static void Main(){
         Console.WriteLine("Name : Hemant Yadu ");
         Console.WriteLine("Class : Mca 3rd sem ");
         Console.Write("Enter a number: ");
         int number = int.Parse(Console.ReadLine());
         if (IsHarshadNumber(number){
            Console.WriteLine($"{number} is a Harshad number.");
         }else{
            Console.WriteLine($"{number} is not a Harshad number.");
         }
        Console.ReadLine();
      }

      static bool IsHarshadNumber(int n) {
         int sumOfDigits = 0;
         int temp = n;
         while (temp > 0) {
            sumOfDigits += temp % 10;
            temp /= 10;
         }
         return n % sumOfDigits == 0;
      }}}
```

Output:



```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
Enter a number: 14526
14526 is a Harshad number.


C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe (process 17840) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the
 console when debugging stops.
Press any key to close this window . . .
```

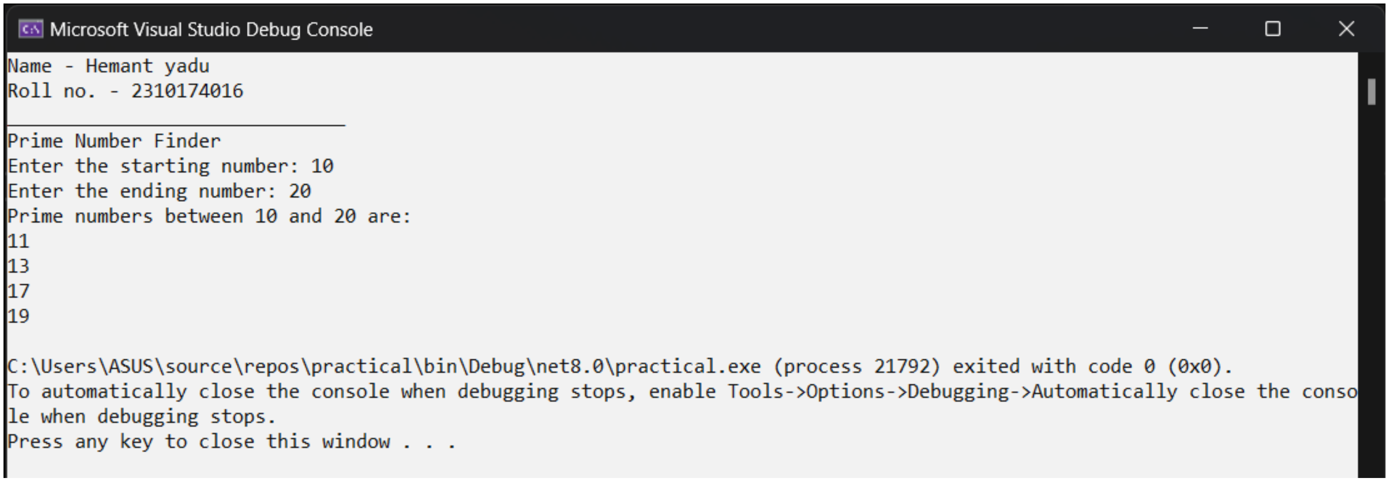
Practical No. 4

Write a C# program to find prime number between range of start number and end number.

Code:

```
using System;
namespace practical{
    internal class prog4{
        static void Main(string[] args) {
            Console.WriteLine("Name : Hemant Yadu");
            Console.WriteLine("Class : Mca 3rd sem");
            Console.WriteLine("Prime Number Finder");
            Console.Write("Enter the starting number: ");
            int start = int.Parse(Console.ReadLine());
            Console.Write("Enter the ending number: ");
            int end = int.Parse(Console.ReadLine());
            Console.WriteLine($"Prime numbers between {start} and {end} are:"
            PrintPrimesInRange(start, end);
            Console.ReadKey();}
        static void PrintPrimesInRange(int start, int end){
            for (int i = start; i <= end; i++) {
                if (IsPrime(i)) {
                    Console.WriteLine(i);
                }
            }
        }
        static bool IsPrime(int number){
            if (number < 2) return false;
            for (int j = 2; j <= Math.Sqrt(number); j++){
                if (number % j == 0) {
                    return false;
                }
            }
            return true;
        }}}
```

Output:



```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
Prime Number Finder
Enter the starting number: 10
Enter the ending number: 20
Prime numbers between 10 and 20 are:
11
13
17
19

C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe (process 21792) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

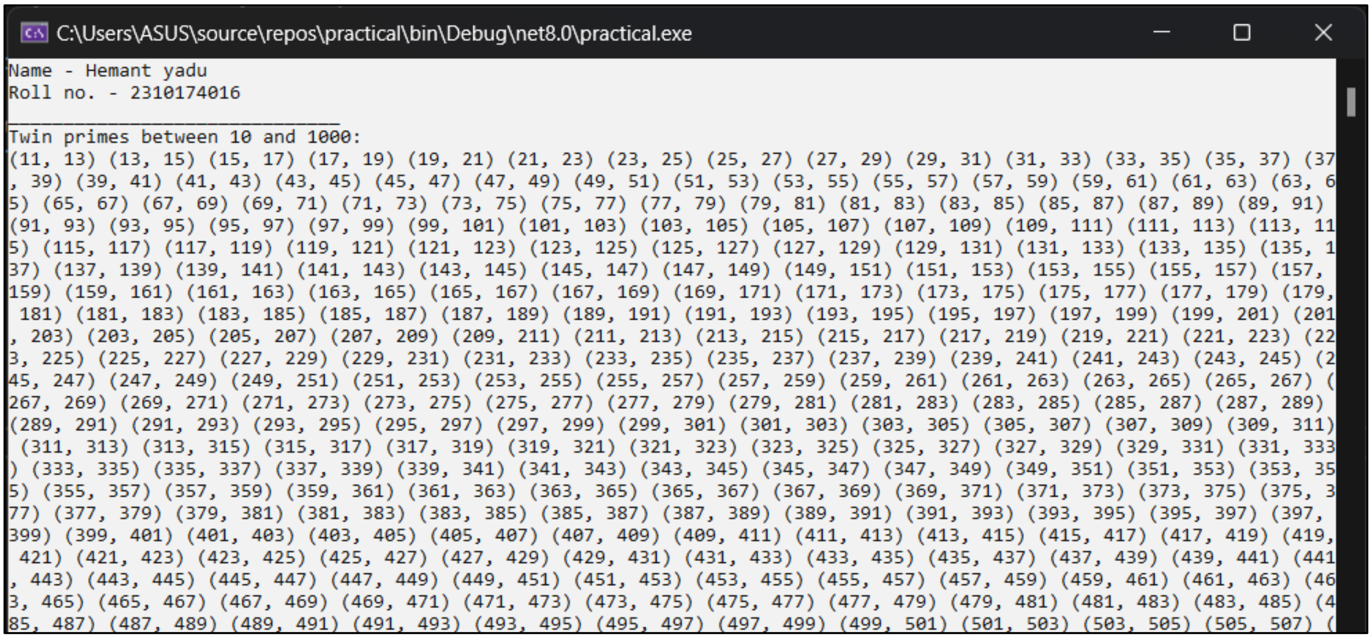
Practical No. 5

Write a C# program to find out tin prime numbers between 10 to 1000. Twin primes are defined to be two consecutive odd number which are prime.

Code:

```
using System;
namespace practical{
    internal class prog5{
        static void Main(string[] args){
            Console.WriteLine("Name: Hemant Yadu");
            Console.WriteLine("Class: MCA 3rd Sem");
            Console.WriteLine("Twin primes between 10 and 1000:");
            for (int i = 11; i < 1000; i += 2 {
                if (IsPrime(i) && IsPrime(i + 2)){
                    Console.WriteLine($"({i}, {i + 2})");
                }
            }
            Console.ReadKey();
        }
        static bool IsPrime(int number) {
            if (number <= 1) return false;
            if (number <= 3) return true;
            if (number % 2 == 0 || number % 3 == 0)
            for (int i = 5; i * i <= number; i += 6) {
                if (number % i == 0 || number % (i + 2) == 0)
            }
            return true;
        }
    }
}
```

Output:


C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe

```
Name - Hemant yadu
Roll no. - 2310174016
__________________________________
Twin primes between 10 and 1000:
(11, 13) (13, 15) (15, 17) (17, 19) (19, 21) (21, 23) (23, 25) (25, 27) (27, 29) (29, 31) (31, 33) (33, 35) (35, 37) (37
, 39) (39, 41) (41, 43) (43, 45) (45, 47) (47, 49) (49, 51) (51, 53) (53, 55) (55, 57) (57, 59) (59, 61) (61, 63) (63, 6
5) (65, 67) (67, 69) (69, 71) (71, 73) (73, 75) (75, 77) (77, 79) (79, 81) (81, 83) (83, 85) (85, 87) (87, 89) (89, 91)
(91, 93) (93, 95) (95, 97) (97, 99) (99, 101) (101, 103) (103, 105) (105, 107) (107, 109) (109, 111) (111, 113) (113, 11
5) (115, 117) (117, 119) (119, 121) (121, 123) (123, 125) (125, 127) (127, 129) (129, 131) (131, 133) (133, 135) (135, 1
37) (137, 139) (139, 141) (141, 143) (143, 145) (145, 147) (147, 149) (149, 151) (151, 153) (153, 155) (155, 157) (157,
159) (159, 161) (161, 163) (163, 165) (165, 167) (167, 169) (169, 171) (171, 173) (173, 175) (175, 177) (177, 179) (179,
 181) (181, 183) (183, 185) (185, 187) (187, 189) (189, 191) (191, 193) (193, 195) (195, 197) (197, 199) (199, 201) (201
, 203) (203, 205) (205, 207) (207, 209) (209, 211) (211, 213) (213, 215) (215, 217) (217, 219) (219, 221) (221, 223) (22
3, 225) (225, 227) (227, 229) (229, 231) (231, 233) (233, 235) (235, 237) (237, 239) (239, 241) (241, 243) (243, 245) (2
45, 247) (247, 249) (249, 251) (251, 253) (253, 255) (255, 257) (257, 259) (259, 261) (261, 263) (263, 265) (265, 267) (
267, 269) (269, 271) (271, 273) (273, 275) (275, 277) (277, 279) (279, 281) (281, 283) (283, 285) (285, 287) (287, 289)
(289, 291) (291, 293) (293, 295) (295, 297) (297, 299) (299, 301) (301, 303) (303, 305) (305, 307) (307, 309) (309, 311)
 (311, 313) (313, 315) (315, 317) (317, 319) (319, 321) (321, 323) (323, 325) (325, 327) (327, 329) (329, 331) (331, 333
) (333, 335) (335, 337) (337, 339) (339, 341) (341, 343) (343, 345) (345, 347) (347, 349) (349, 351) (351, 353) (353, 35
5) (355, 357) (357, 359) (359, 361) (361, 363) (363, 365) (365, 367) (367, 369) (369, 371) (371, 373) (373, 375) (375, 3
77) (377, 379) (379, 381) (381, 383) (383, 385) (385, 387) (387, 389) (389, 391) (391, 393) (393, 395) (395, 397) (397,
399) (399, 401) (401, 403) (403, 405) (405, 407) (407, 409) (409, 411) (411, 413) (413, 415) (415, 417) (417, 419) (419,
 421) (421, 423) (423, 425) (425, 427) (427, 429) (429, 431) (431, 433) (433, 435) (435, 437) (437, 439) (439, 441) (441
, 443) (443, 445) (445, 447) (447, 449) (449, 451) (451, 453) (453, 455) (455, 457) (457, 459) (459, 461) (461, 463) (46
3, 465) (465, 467) (467, 469) (469, 471) (471, 473) (473, 475) (475, 477) (477, 479) (479, 481) (481, 483) (483, 485) (4
85, 487) (487, 489) (489, 491) (491, 493) (493, 495) (495, 497) (497, 499) (499, 501) (501, 503) (503, 505) (505, 507) (
```

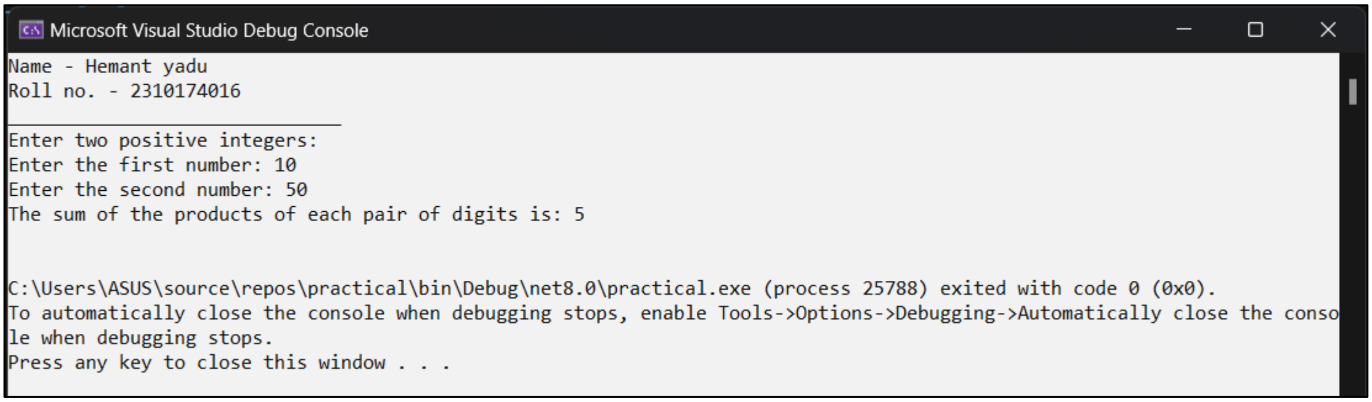
Practical No. 6

Enter two positive integers and calculate and print out the sum of the products of each pair of digits occupying the same position in the two numbers.

Code:

```
using System;
namespace practical{
    internal class prog6{
        static void Main(string[] args){
            Console.WriteLine("Name : Hemant Yadu");
            Console.WriteLine("Class : Mca 3rd sem");
            Console.WriteLine("Enter two positive integers:");
            Console.Write("Enter the first number: ");
            string num1 = Console.ReadLine();

            Console.Write("Enter the second number: ");
            string num2 = Console.ReadLine();

            int maxLength = Math.Max(num1.Length, num2.Length);
            num1 = num1.PadLeft(maxLength, '0');
            num2 = num2.PadLeft(maxLength, '0');
            int sumOfProducts = 0;
            for (int i = 0; i < maxLength; i++){
                // Convert character to integer and calculate the product
                int digit1 = num1[i] - '0';
                int digit2 = num2[i] - '0';
                sumOfProducts += digit1 * digit2;
            }
            Console.WriteLine($"The sum of the products of each pair of digits is: {sumOfProducts}");
            Console.ReadLine();
        }
    }
}
```

Output:



```
Name - Hemant yadu
Roll no. - 2310174016
_______________________________________
Enter two positive integers:
Enter the first number: 10
Enter the second number: 50
The sum of the products of each pair of digits is: 5


C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe (process 25788) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Practical No. 7

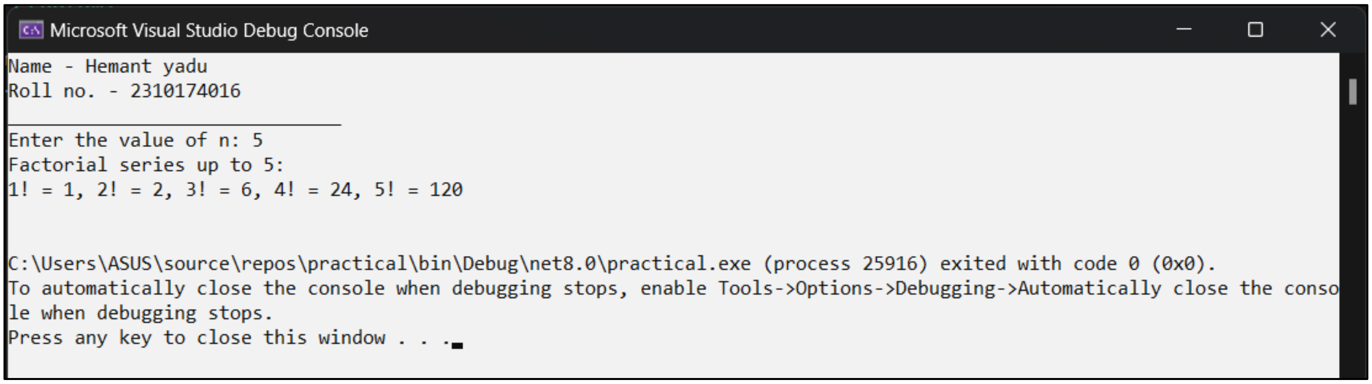
Write a C# program to display factorial series up to nth term

1! 2! 3! 4! …… n!

Code:

```
using System;
namespace practical{
    internal class prog7{
        static void Main(){
            Console.WriteLine("Name : Hemant Yadu");
            Console.WriteLine("Class : Mca 3rd sem");
            Console.Write("Enter the value of n: ");
            int n = int.Parse(Console.ReadLine());
            Console.WriteLine("Factorial series up to {0}:", n);
            for (int i = 1; i <= n; i++)  {
                Console.Write("{0}! = {1}", i, Factorial(i));
                if (i < n)  {
                    Console.Write(", ");
                }
            }
            Console.WriteLine(); // Move to next line after the series
            Console.ReadLine();
        }
        static long Factorial(int number) {
            long result = 1;
            for (int i = 1; i <= number; i++) {
                result *= i;
            }
            return result;
        }
    }
}
```

Output:



```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
Enter the value of n: 5
Factorial series up to 5:
1! = 1, 2! = 2, 3! = 6, 4! = 24, 5! = 120


C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe (process 25916) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

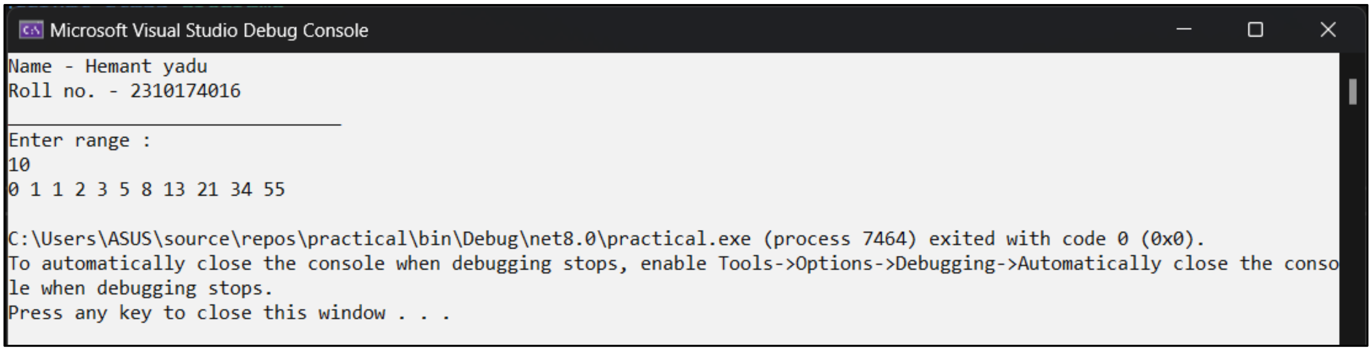
Practical No. 8

Write the program that display Fibonacci series up to nth term.
0 1 1 2 3 5 8 13 ……n

Code:

```
using System;
namespace practical{
    internal class prog8 {
        static void Main(string[] args){
            int num1 = 0, num2 = 1, num3, range;
            Console.WriteLine("Name : Hemant
            Yadu"); Console.WriteLine("Class : Mca
            3rd sem");Console.WriteLine("Enter range
            : ");
            range = Convert.ToInt32(Console.ReadLine());
            Console.Write(num1 + " ");
            Console.Write(num2 + " ");
            while (range > 1){
                num3 = num1 + num2;
                Console.Write(num3 + " ");
                num1 = num2;
                num2 = num3;
                range--;
            }
            Console.ReadLine();
        }
    }
}
```

Output:



```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
Enter range :
10
0 1 1 2 3 5 8 13 21 34 55

C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe (process 7464) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Practical No. 9

Write a C# program to do the following operations:

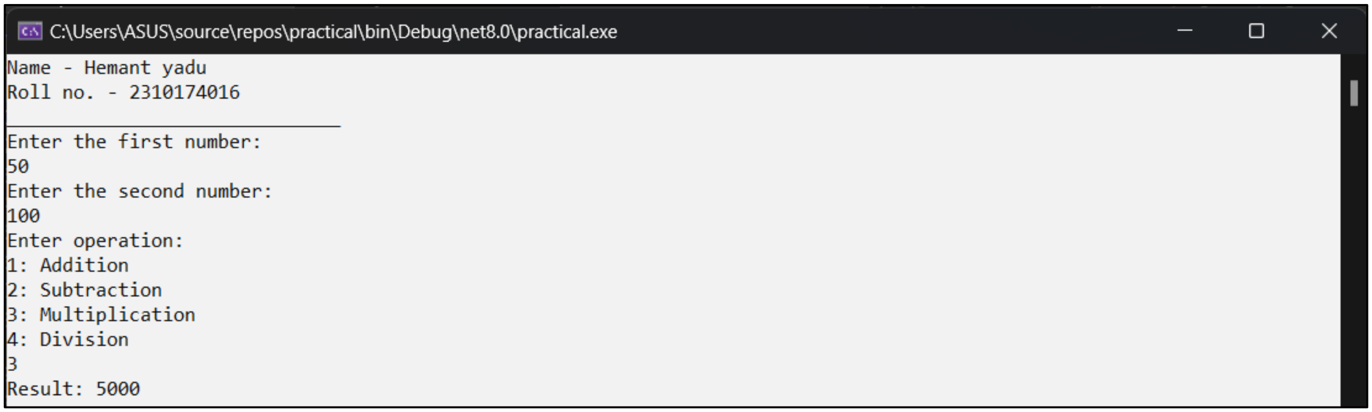
- **Read any two positive integer operands (say op1 & op2) and one character type operator (say opr). Note that opr is any mathematical operator.**
- **Depending upon the operator, do the appropriate operation.**

Code:

```
using System;
namespace practical{
    internal class prog9{
        static void Main(string[] args){
            int opr ,  int op1;
            int op2 , int result;
            Console.WriteLine("Name : Hemant
            Yadu"); Console.WriteLine("Class: Mca
            3rd sem");
            Console.WriteLine("Enter the first number: ");
            op1 = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter the second number: ");
            op2 = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter operation: ");
            Console.WriteLine("1: Addition");
            Console.WriteLine("2: Subtraction");
            Console.WriteLine("3: Multiplication");
            Console.WriteLine("4: Division");
            opr = int.Parse(Console.ReadLine());

            switch (opr){
                case 1:
                    result = op1 + op2;
                    break;
                case 2:
                    result = op1 - op2;
                    break;
                case 3:
                    result = op1 * op2;
                    break;
                case 4:
                    if (op2 != 0) // Check for division by zero
                    {
                        result = op1 / op2;
                    }
                    else
                    {
                        Console.WriteLine("Cannot divide by zero!");
                        return; // Exit the program
                    }
```

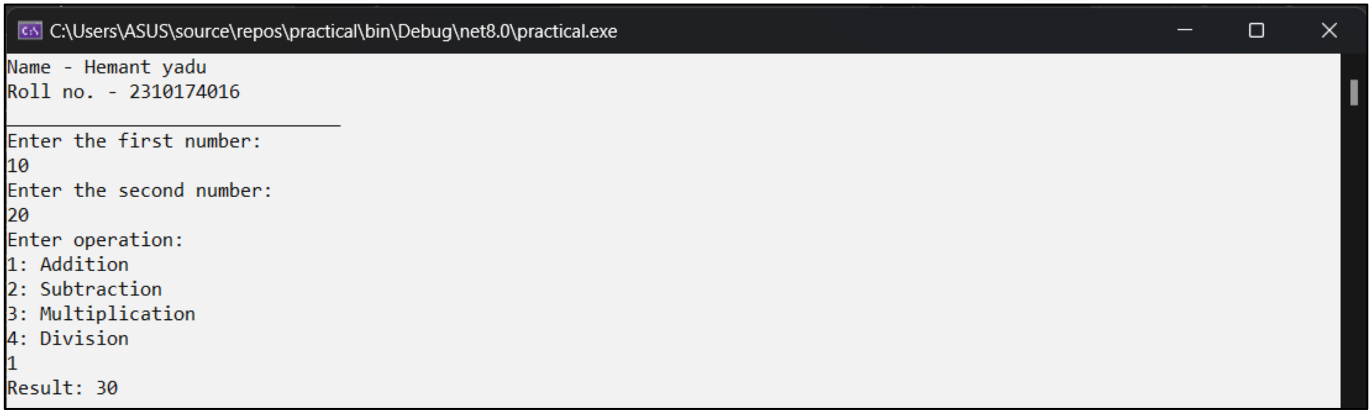
```
                        break;
                    default:
Console.WriteLine("Invalid choice");return; // Exit the
program
                }

                Console.WriteLine("Result: {0}", result);
                Console.ReadLine(); // Wait for user input before closing
            }
        }

    }
```

Output:



C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe

```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
Enter the first number:
50
Enter the second number:
100
Enter operation:
1: Addition
2: Subtraction
3: Multiplication
4: Division
3
Result: 5000
```



C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe

```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
Enter the first number:
10
Enter the second number:
20
Enter operation:
1: Addition
2: Subtraction
3: Multiplication
4: Division
1
Result: 30
```

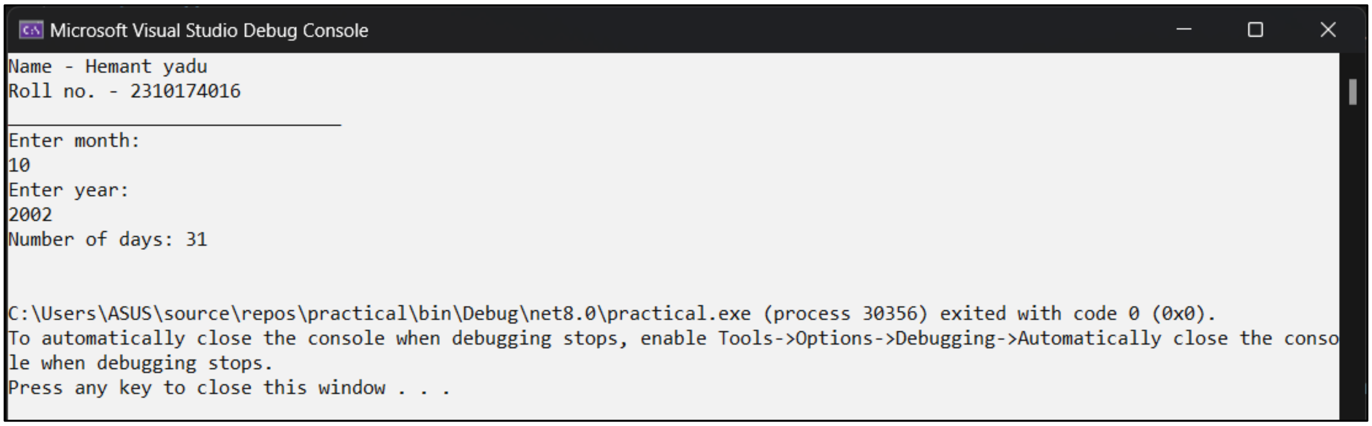
Practical No. 10

Write a C# program to read month (1-12) and 4digited year and display total days in given month and year.

Code:

```
using System;
namespace practical{
    internal class prog10 {
        static void Main(string[] args)  {
            int month , int year;
            Console.WriteLine("Name : Hemant
            Yadu"); Console.WriteLine("Class : Mca
            3rd sem");Console.WriteLine("Enter
            month: "); month =
            int.Parse(Console.ReadLine());
            Console.WriteLine("Enter year: ");
            year = int.Parse(Console.ReadLine());
            if (month == 2) {
                if (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0)) {
                    Console.WriteLine("Number of days: 29");
                } else {
                    Console.WriteLine("Number of days: 28");
                }
            }
            else if (month == 1 || month == 3 || month == 5 || month == 7 ||
                     month == 8 || month == 10 || month == 12) {
                Console.WriteLine("Number of days: 31");2
            }else{
                Console.WriteLine("Number of days: 30");
            }
            Console.ReadLine();
        }
    }
}
```

Output:



```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
Enter month:
10
Enter year:
2002
Number of days: 31


C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe (process 30356) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Practical No. 11

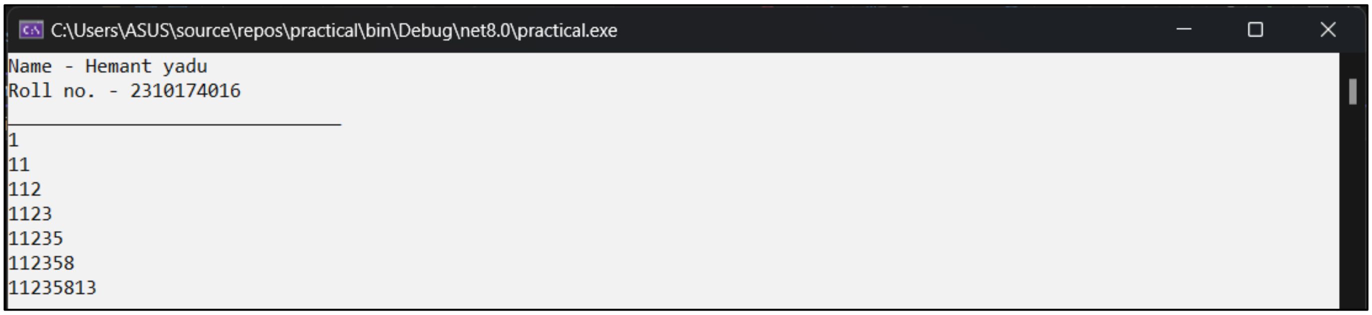
Write a C# program that display following pattern. N=7

```
 1
 1 1
1 1 2
 1 1 2 3
1 1 2 3 5
1 1 2 3 5 8
```

Code:

```
using System;
namespace practical{
   internal class prog11 {
      static void Main(string[] args){
         int N = 7;
         int[] fib = new int[N];
         Console.WriteLine("Name : Hemant
         Yadu"); Console.WriteLine("Class : Mca
         3rd sem");fib[0] = 1;
         if (N > 1) {
            fib[1] = 1;
            for (int i = 2; i < N; i++){
               fib[i] = fib[i - 1] + fib[i - 2];
            }
         }
           for(int i = 1; i<=N; i++) {
             for(int j = 0; j < i; j++){
               Console.Write(fib[j]);
            }
            Console.WriteLine();
            Console.ReadLine();
         }}}}
```

Output:


C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe

```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
1
11
112
1123
11235
112358
11235813
```

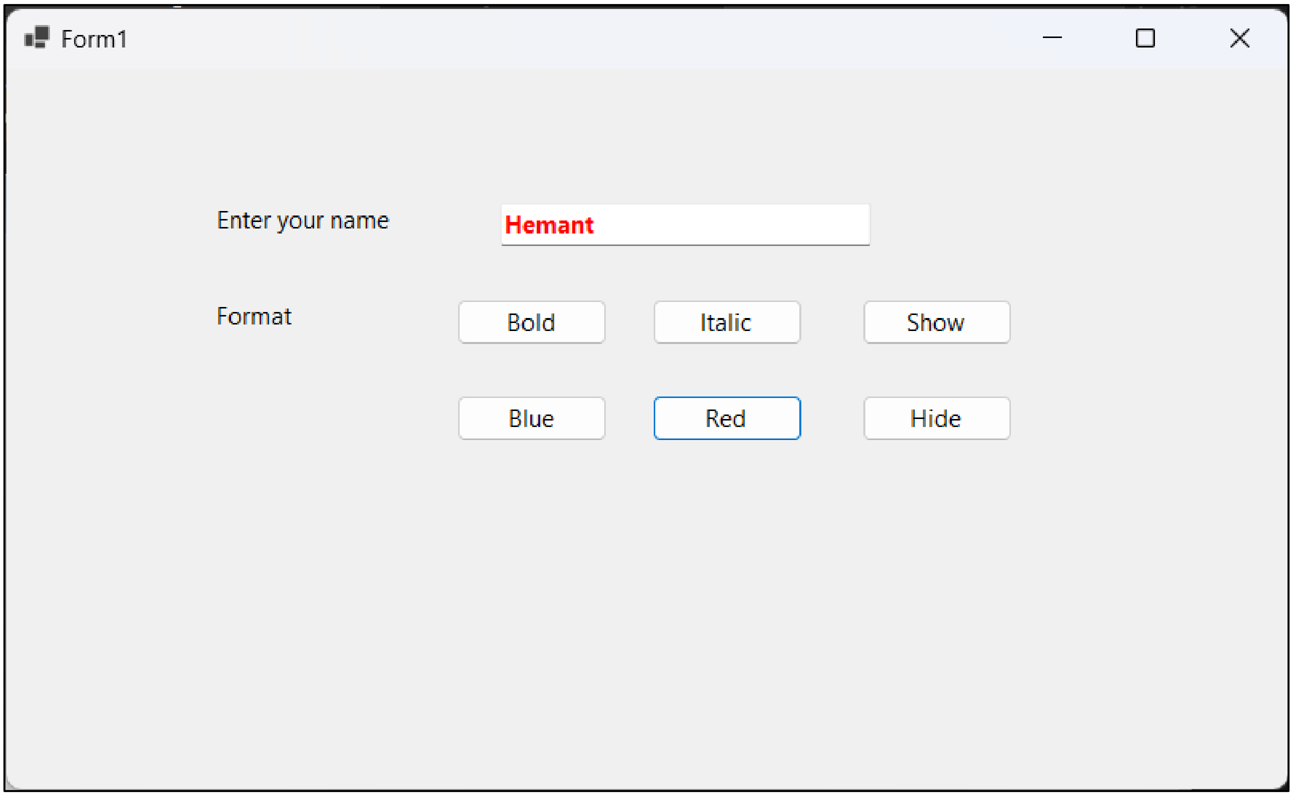
Practical No. 12

Design the form having 1 Text box and buttons like Bold, Italic, Red & Blue, show and hide When user presses an y of this command buttons then the content of text box will be changed to red or blue or bold or italic and Clicking on the "Show" button it will display a Textbox & when user clicks on "Hide" button then it will remove the Textbox. (Use Visible Property).

Code:

```
namespace Form1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            textBox1.Font = new Font(textBox1.Font, FontStyle.Bold);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            textBox1.Font = new Font(textBox1.Font, FontStyle.Italic);
        }

        private void button4_Click(object sender, EventArgs e)
        {
            textBox1.ForeColor = Color.Blue;
        }

        private void button5_Click(object sender, EventArgs e)
        {
            textBox1.ForeColor = Color.Red;
        }

        private void button3_Click(object sender, EventArgs e)
        {
            textBox1.Visible = true;
        }

        private void button6_Click(object sender, EventArgs e)
        {
            textBox1.Visible = false;
        }
    }
}
```

Form :


Form1
Enter your name
Hemant
Format
Bold
Italic
Show
Blue
Red
Hide

Practical No. 13

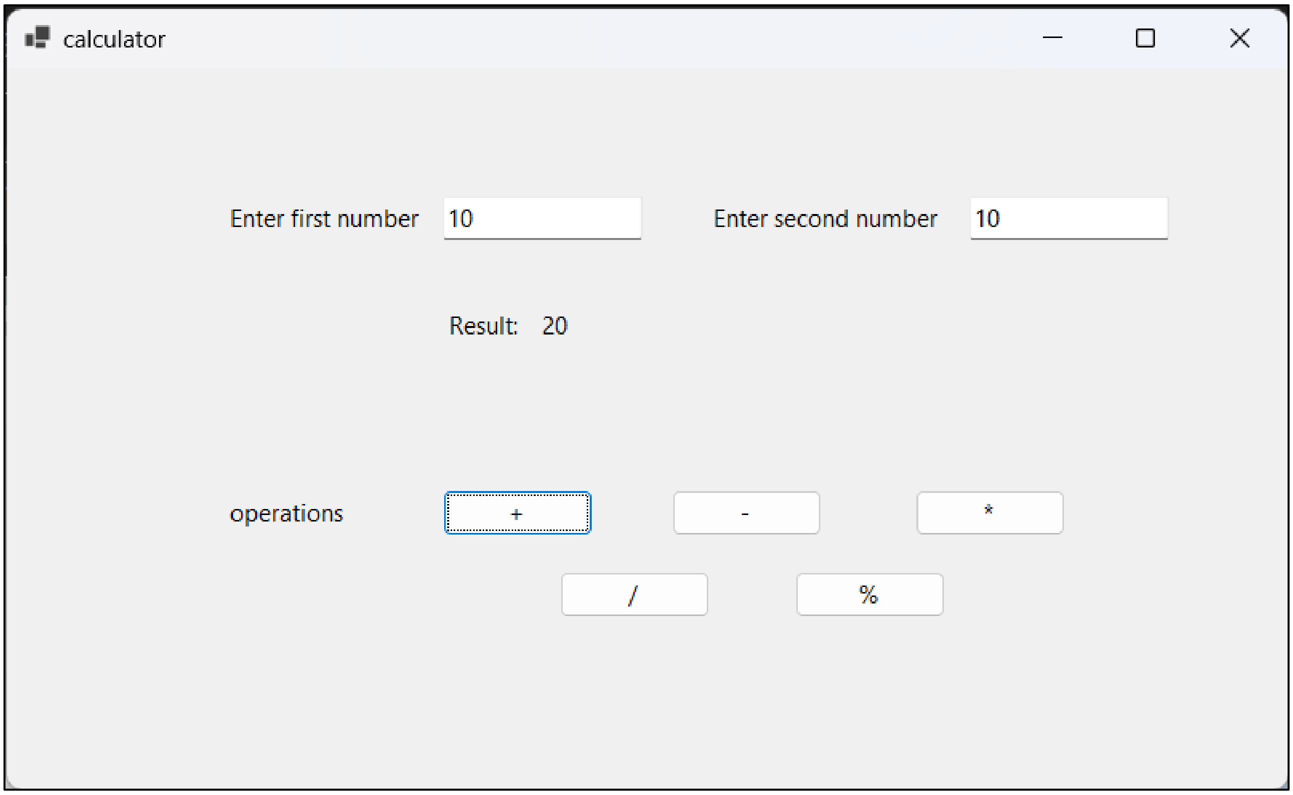
Design the form that calculates Sum, Multiplication, Division and Subtraction of two numbers.

Code:

```
namespace Form2

{

   public partial class Form1 : Form

   {

      public Form1()

      {

         InitializeComponent();

      }

      private void Add_Click(object sender, EventArgs e)

      {

         int num1 = Convert.ToInt32(number1.Text);
         int num2 = Convert.ToInt32(number2.Text);
         int sum = num1 + num2;

         label5.Text = sum.ToString();

      }


      private void Sub_Click(object sender, EventArgs e)

      {

         int num1 = Convert.ToInt32(number1.Text);
         int num2 = Convert.ToInt32(number2.Text);
         int sub = num1 - num2;

         label5.Text = sub.ToString();

      }

      private void Mul_Click(object sender, EventArgs e)

      {

         int num1 = Convert.ToInt32(number1.Text);
         int num2 = Convert.ToInt32(number2.Text);
         int mul = num1 * num2;

         label5.Text = mul.ToString();

      }

      private void Div_Click(object sender, EventArgs e)
```

```
        {
            int num1 = Convert.ToInt32(number1.Text);
            int num2 = Convert.ToInt32(number2.Text);
            int div = num1 / num2;

            label5.Text = mul.ToString();

        }
        private void Mod_Click(object sender, EventArgs e)

        {
            int num1 = Convert.ToInt32(number1.Text);
            int num2 = Convert.ToInt32(number2.Text);
            int mod = num1 % num2;

            label5.Text = mod.ToString();

        }

    }

}
```

Calculator:



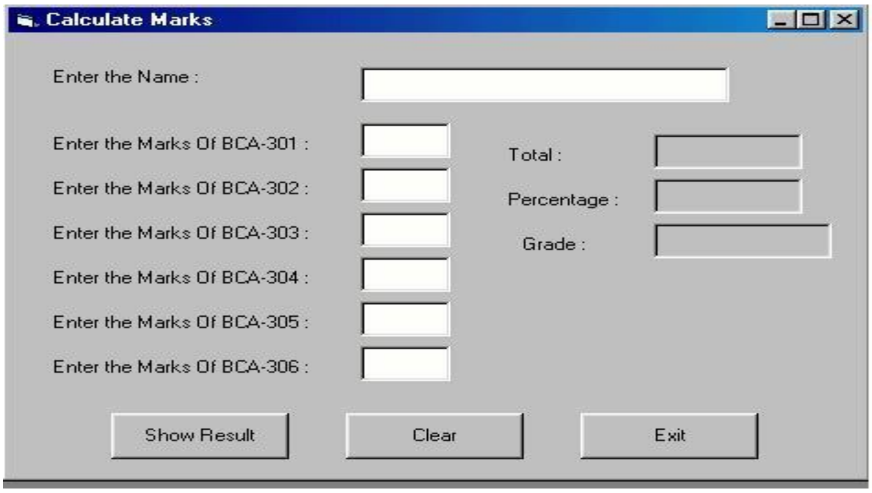
Enter first number 10

Enter second number 10

Result: 20

Practical No. 14

Design the following form.

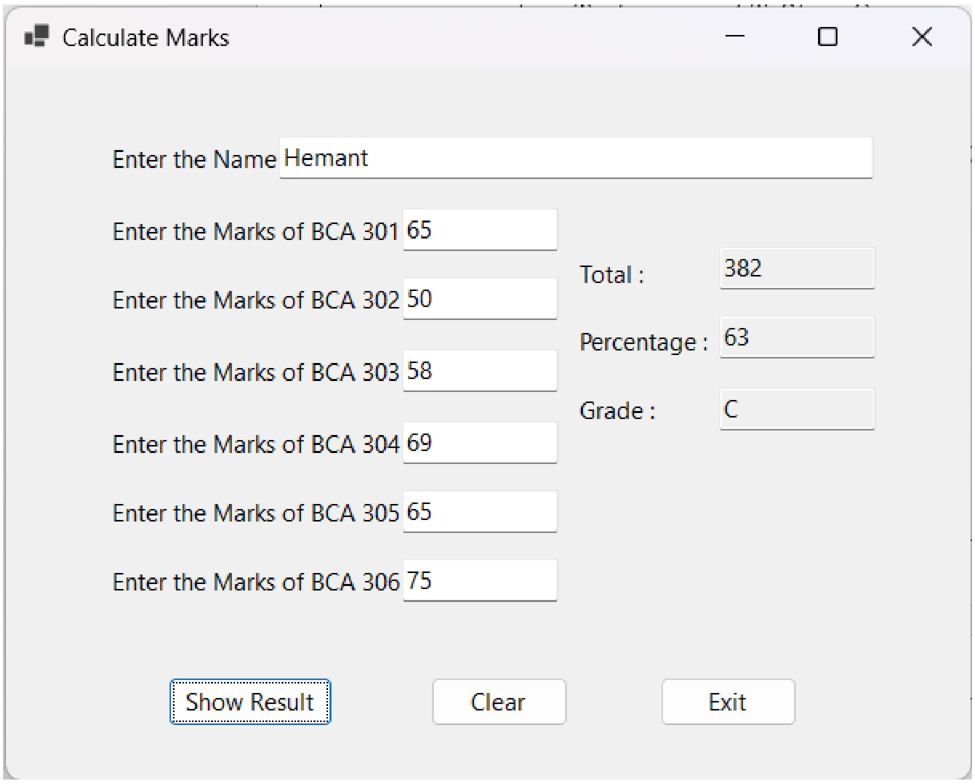

Calculate Marks
Enter the Name :
Enter the Marks Of BCA-301 :
Enter the Marks Of BCA-302 :
Enter the Marks Of BCA-303 :
Enter the Marks Of BCA-304 :
Enter the Marks Of BCA-305 :
Enter the Marks Of BCA-306 :
Total :
Percentage :
Grade :
Show Result
Clear
Exit

Code:

```
namespace C_List2
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
         name.KeyPress += new KeyPressEventHandler(name_KeyPress);
         bca1.KeyPress += new KeyPressEventHandler(bca1_KeyPress);
         bca2.KeyPress += new KeyPressEventHandler(bca2_KeyPress);
         bca3.KeyPress += new KeyPressEventHandler(bca3_KeyPress);
         bca4.KeyPress += new KeyPressEventHandler(bca4_KeyPress);
         bca5.KeyPress += new KeyPressEventHandler(bca5_KeyPress);
         bca6.KeyPress += new KeyPressEventHandler(bca6_KeyPress);
         bca1.TextChanged += new EventHandler(bca1_TextChanged);
         bca2.TextChanged += new EventHandler(bca2_TextChanged);
         bca3.TextChanged += new EventHandler(bca3_TextChanged);
         bca4.TextChanged += new EventHandler(bca4_TextChanged);
         bca5.TextChanged += new EventHandler(bca5_TextChanged);
         bca6.TextChanged += new EventHandler(bca6_TextChanged);
      }

      //validating form.
      private void name_KeyPress(object sender, KeyPressEventArgs e) {
         if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar)){
            e.Handled = true;
         }
      }
      private void name_TextChanged(object sender, EventArgs e){
         if (!System.Text.RegularExpressions.Regex.IsMatch(name.Text, @"^[a-zA-Z]*$")) {
            MessageBox.Show("Please Enter Valid Name.");
            name.Text = "";
         }
      }

      private void bca1_KeyPress(object sender, KeyPressEventArgs e){
         if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)){
            e.Handled = true;
```

```
    }
}
private void bca1_TextChanged(object sender, EventArgs e) {
    if(int.TryParse(bca1.Text, out int value)) {
        if (value > 100) {
            MessageBox.Show("Please Enter Positive Number less than or equal to 100");
        }
    }
}
private void bca2_KeyPress(object sender, KeyPressEventArgs e){
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)){
        e.Handled = true;
    }
}
private void bca2_TextChanged(object sender, EventArgs e){
    if (int.TryParse(bca2.Text, out int value)){
        if (value > 100){
            MessageBox.Show("Please Enter Positive Number less than or equal to 100");
        }
    }
}
private void bca3_KeyPress(object sender, KeyPressEventArgs e){
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)){
        e.Handled = true;
    }
}
private void bca3_TextChanged(object sender, EventArgs e){
    if (int.TryParse(bca3.Text, out int value)){
        if (value > 100){
            MessageBox.Show("Please Enter Positive Number less than or equal to 100");
        }
    }
}
private void bca4_KeyPress(object sender, KeyPressEventArgs e){
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)){
        e.Handled = true;
    }
}
private void bca4_TextChanged(object sender, EventArgs e){
    if (int.TryParse(bca4.Text, out int value))
    {
        if (value > 100){
            MessageBox.Show("Please Enter Positive Number less than or equal to 100");
        }
    }
}
private void bca5_KeyPress(object sender, KeyPressEventArgs e){
    if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)){
        e.Handled = true;
    }
}
private void bca5_TextChanged(object sender, EventArgs e){
    if (int.TryParse(bca5.Text, out int value)){
        if (value > 100){
            MessageBox.Show("Please Enter Positive Number less than or equal to 100");
        }
    }
}
private void bca6_KeyPress(object sender, KeyPressEventArgs e){
        if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar)){
            e.Handled = true;
        }
    }
    private void bca6_TextChanged(object sender, EventArgs e){
        if (int.TryParse(bca6.Text, out int value)){
            if (value > 100){
                MessageBox.Show("Please Enter Positive Number less than or equal to 100");
            }
        }
    }

    private void result_Click(object sender, EventArgs e){
        int num1, num2, num3, num4, num5, num6;
        num1 = Convert.ToInt32(bca1.Text);
        num2 = Convert.ToInt32(bca2.Text);
        num3 = Convert.ToInt32(bca3.Text);
        num4 = Convert.ToInt32(bca4.Text);
        num5 = Convert.ToInt32(bca5.Text);
        num6 = Convert.ToInt32(bca6.Text);

        int Total = num1 + num2 + num3 + num4 + num5 + num6;
        double percent = Total / 6;
        String grade;
        if (percent >= 80){
            grade = "A";
        }
        else if (percent >= 70){
            grade = "B";
        }
        else if (percent >= 60){
            grade = "C";
        }
        else if (percent >= 50){
            grade = "D";
        }
        else if (percent >= 40){
            grade = "E";
        }
        else{
            grade = "F";
        }

        total.Text = Total.ToString();
        percentage.Text = percent.ToString();
        grades.Text = grade.ToString();
    }

    private void clear_Click(object sender, EventArgs e){
        name.Text = string.Empty;
        bca1.Text = string.Empty;
        bca2.Text = string.Empty;
        bca3.Text = string.Empty;
        bca4.Text = string.Empty;
        bca5.Text = string.Empty;
        bca6.Text = string.Empty;
        total.Text = string.Empty;
        percentage.Text = string.Empty;
        grades.Text = string.Empty;
        }

        private void exit_Click(object sender, EventArgs e){
            this.Close();
        }
    }
}
```

Output:



Enter the Name Hemant

Enter the Marks of BCA 301 65

Enter the Marks of BCA 302 50

Enter the Marks of BCA 303 58

Enter the Marks of BCA 304 69

Enter the Marks of BCA 305 65

Enter the Marks of BCA 306 75

Total : 382

Percentage : 63

Grade : C

Practical No. 15

Design a form to create conversion calculator like

- **Decimal number to Binary, Octal & Hexadecimal.**
- **Binary number to Decimal, Octal & Hexadecimal.**
- **Octal number to hexa, binary ,decimal etc.**

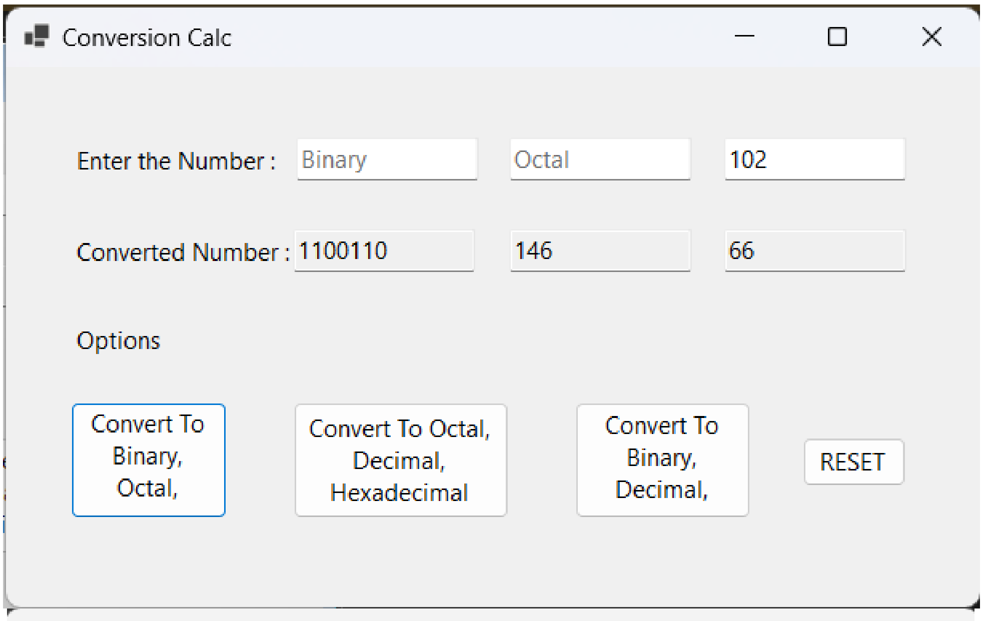
Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace C_List2
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
            binary.KeyPress += new KeyPressEventHandler(binary_KeyPress);
            octal.KeyPress += new KeyPressEventHandler(octal_KeyPress);
            deci.KeyPress += new KeyPressEventHandler(deci_KeyPress);
            binary.TextChanged += new EventHandler(binary_TextChanged);
            octal.TextChanged += new EventHandler(octal_TextChanged);
            deci.TextChanged += new EventHandler(deci_TextChanged);
        }
        private void binary_KeyPress(object Sender, KeyPressEventArgs e)
        {
            if (!char.IsControl(e.KeyChar) && e.KeyChar != '0' && e.KeyChar != '1')
            {
                e.Handled = true; // Ignore any key that isn't 0, 1, or control (backspace, delete)
            }
        }
        private void binary_TextChanged(object Sender, EventArgs e)
        {
            if (int.TryParse(binary.Text, out int value))
            {
                if (value < 0 )
                {
                    MessageBox.Show("Pls Enter Valid Numbers.");
                }
            }
            octal.Clear();
            deci.Clear();

        }
        private void octal_KeyPress(object Sender, KeyPressEventArgs e)
        {
            if (!char.IsControl(e.KeyChar) && (e.KeyChar < '0' || e.KeyChar > '7'))
            {
                e.Handled = true; // Ignore any key that isn't 0-7 or control
            }
```

```
}
private void octal_TextChanged(object Sender, EventArgs e)
{
  if (int.TryParse(octal.Text, out int value))
  {
    if (value < 0 )
    {
      MessageBox.Show("Pls Enter Valid Numbers.");
    }
  }
  binary.Clear();
  deci.Clear();

}
private void deci_KeyPress(object Sender, KeyPressEventArgs e)
{
  if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
  {
    e.Handled = true;
  }
}
private void deci_TextChanged(object Sender, EventArgs e)
{
  if (int.TryParse(deci.Text, out int value))
  {
    if (value < 0)
    {
      MessageBox.Show("Pls Enter Valid Numbers.");
    }
    binary.Clear();
    octal.Clear();
  }

}

private void button1_Click(object sender, EventArgs e)
{
  if (string.IsNullOrWhiteSpace(deci.Text))
  {
    MessageBox.Show("Please enter a decimal value before performing this action.");
    return; // Exit the method without performing any further action
  }
  else {
    //converting
    int number1 = int.Parse(deci.Text);
    string binary = Convert.ToString(number1, 2);
    string octal = Convert.ToString(number1, 8);
    string hex = number1.ToString("X");
    //assigning values.
    result1.Text = Convert.ToString(binary);
    result2.Text = Convert.ToString(octal);
    result3.Text = Convert.ToString(hex);
  }
}

private void button2_Click(object sender, EventArgs e)
{
  if (string.IsNullOrWhiteSpace(binary.Text))
```

```
          {
            MessageBox.Show("Please enter a binary value before performing this action.");
            return; // Exit the method without performing any further action
          }
          else {
            //first convert to binary.
            int number1 = Convert.ToInt32(binary.Text, 2);
            //converting
            string octal = Convert.ToString(number1, 8);
            string deci = Convert.ToString(number1, 10);
            string hex = number1.ToString("X");
            //assigning values.
            result1.Text = Convert.ToString(octal);
            result2.Text = Convert.ToString(deci);
            result3.Text = Convert.ToString(hex);
          }


        }

        private void button3_Click(object sender, EventArgs e)
        {

          if (string.IsNullOrWhiteSpace(octal.Text))
          {
            MessageBox.Show("Please enter a octal value before performing this action.");
            return; // Exit the method without performing any further action
          }
          else
          {
            int number1 = Convert.ToInt32(octal.Text, 8);
            //converting
            string binary = Convert.ToString(number1, 2);
            string deci = Convert.ToString(number1, 10);
            string hexa = Convert.ToString(number1, 16);
            //assigning values.
            result1.Text = Convert.ToString(binary);
            result2.Text = Convert.ToString(deci);
            result3.Text = Convert.ToString(hexa);
          }
        }
        private void button4_Click(object sender, EventArgs e)
        {
          binary.Clear();
          octal.Clear();
          deci.Clear();
          result1.Text = "";
          result2.Text = "";
          result3.Text = "";
        }
    }
}
```

Output:



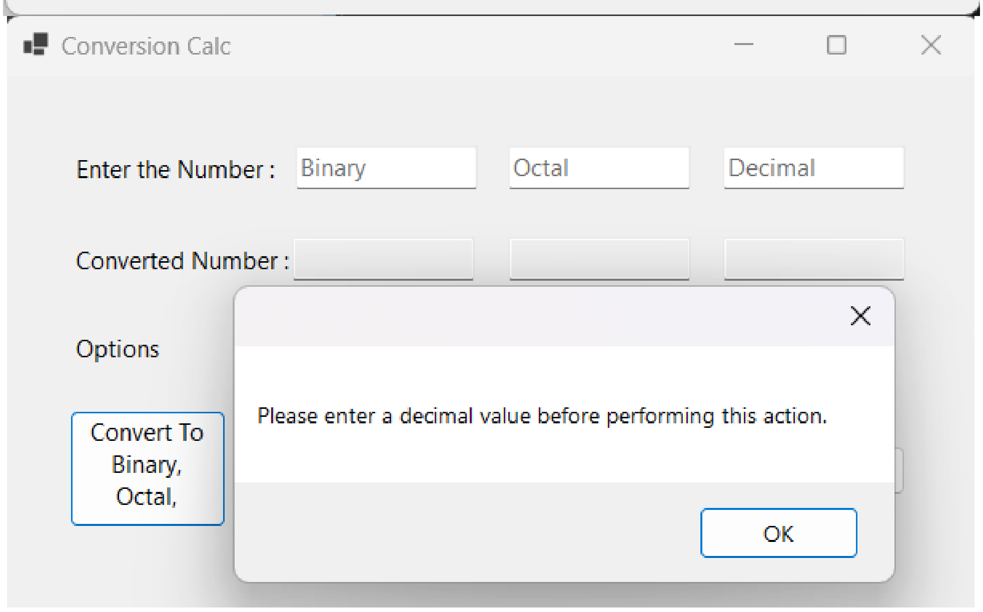
Conversion Calc

Enter the Number : Binary Octal 102

Converted Number : 1100110 146 66

Options

Convert To Binary, Octal,
Convert To Octal, Decimal, Hexadecimal
Convert To Binary, Decimal,
RESET



Conversion Calc

Enter the Number : Binary Octal Decimal

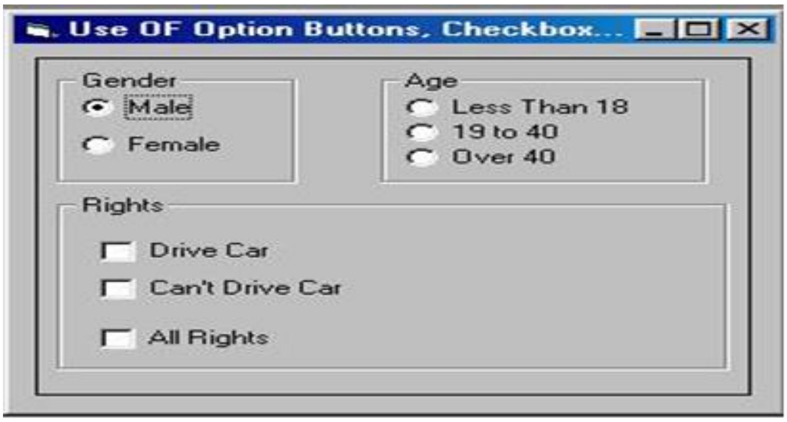
Converted Number :

Options

Please enter a decimal value before performing this action.

Practical No. 16

Design the following form. So when user clicks on Radio Button then select appropriate checkbox.

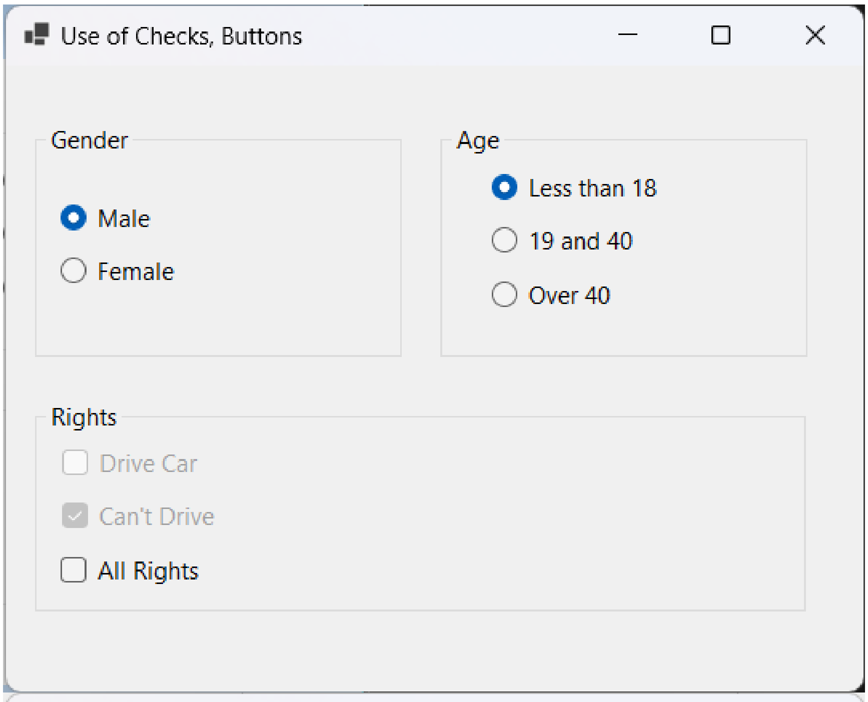


Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace C_List2
{
   public partial class Form3 : Form
   {
      public Form3()
      {
         InitializeComponent();
      }

      private void male_CheckedChanged(object sender, EventArgs e)
      {
         if (male.Checked)
            female.Checked = false;
      }

      private void female_CheckedChanged(object sender, EventArgs e)
      {
         if (female.Checked)
            male.Checked = false;
      }

      private void ls18_CheckedChanged(object sender, EventArgs e)
      {
         if (ls18.Checked && (male.Checked || female.Checked))
         {
            allrights.Enabled = true;
            ls1940.Checked = false;
            ov40.Checked = false;
            nodrive.Checked = true;
```

```
                allrights.Checked = false;
                drive.Checked = false;
                nodrive.Enabled = false;
                drive.Enabled = false;
            }
        }

        private void ls1940_CheckedChanged(object sender, EventArgs e)
        {
            if (ls1940.Checked && (male.Checked || female.Checked))
            {
                drive.Enabled = true;
                nodrive.Enabled = true;
                ls18.Checked = false;
                ov40.Checked = false;
                allrights.Checked = true;
                allrights.Enabled = false;

            }
        }

        private void ov40_CheckedChanged(object sender, EventArgs e)
        {
            if (ov40.Checked && (male.Checked || female.Checked))
            {
                drive.Enabled = true;
                nodrive.Enabled = true;
                ls18.Checked = false;
                ls1940.Checked = false;
                allrights.Checked = true;
                allrights.Enabled = false;

            }
        }

        private void drive_CheckedChanged(object sender, EventArgs e)
        {
            if (drive.Checked)
                nodrive.Checked = false;
        }

        private void nodrive_CheckedChanged(object sender, EventArgs e)
        {
            if (nodrive.Checked)
                drive.Checked = false;
        }
    }
}
```

Output:



Use of Checks, Buttons

Gender

● Male

○ Female

Age

● Less than 18

○ 19 and 40

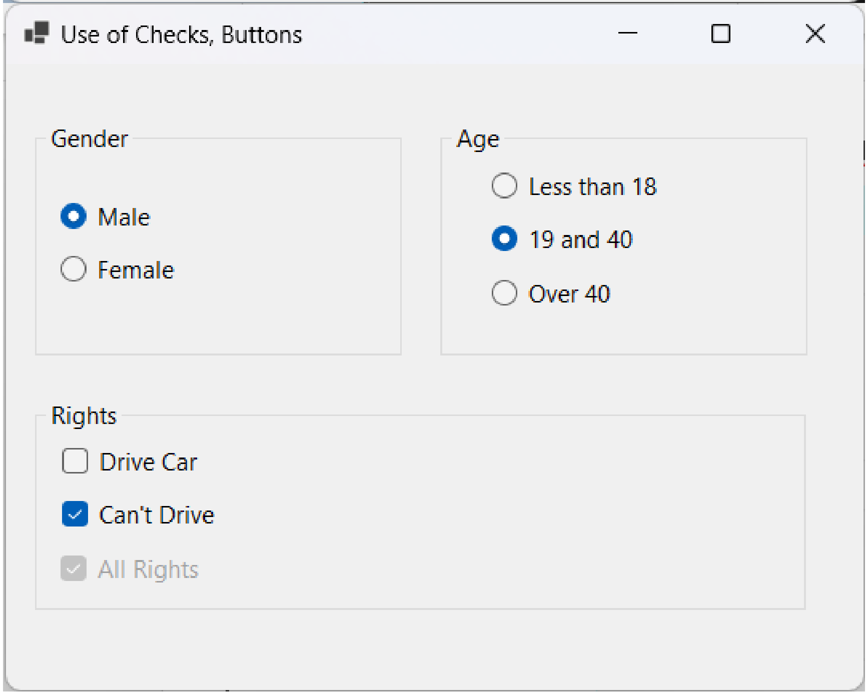
○ Over 40

Rights

□ Drive Car

☑ Can't Drive

□ All Rights



Use of Checks, Buttons

Gender

● Male

○ Female

Age

○ Less than 18

● 19 and 40

○ Over 40

Rights

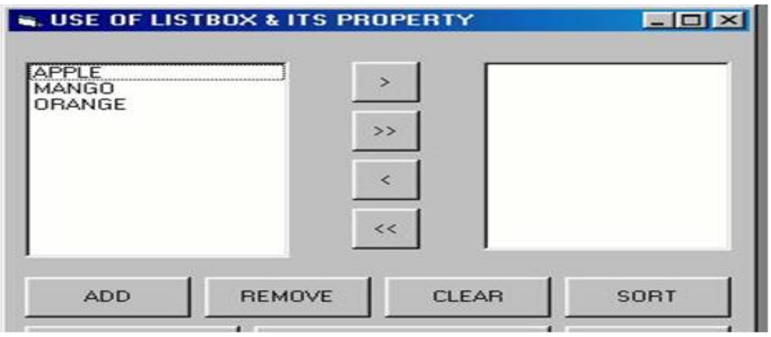
□ Drive Car

☑ Can't Drive

☑ All Rights

Practical No. 17

program shows the functionality of listbox:



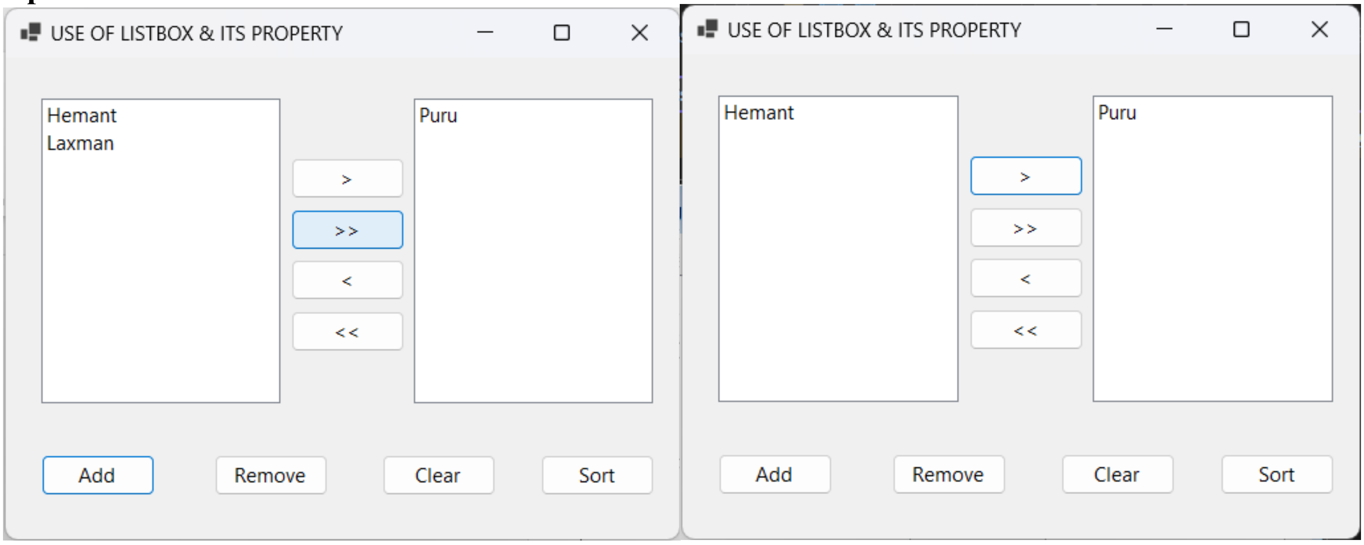
Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using static System.Net.Mime.MediaTypeNames;

namespace C_List2
{
   public partial class Form4 : Form{
      string SelectedItem;
      Boolean sortFlag = true;
      public Form4(){
         InitializeComponent();
         listView1.View = View.List;
         listView2.View = View.List;
         listView1.MultiSelect = false;
      }
      private void button5_Click(object sender, EventArgs e){
         using (TextInput Dialog = new TextInput()){
            if (Dialog.ShowDialog() == DialogResult.OK){
               string list = Dialog.InputText;
               if (!string.IsNullOrEmpty(list)){
                  listView1.Items.Add(new ListViewItem(list));
               }
            }

         }
      }
      private void button6_Click(object sender, EventArgs e){
         // Check if there is any selected item
         if (listView1.SelectedItems.Count > 0){
            // Remove the selected item
            listView1.Items.Remove(listView1.SelectedItems[0]);
         }
         else{
            MessageBox.Show("No item selected to remove.");
         }
      }
```

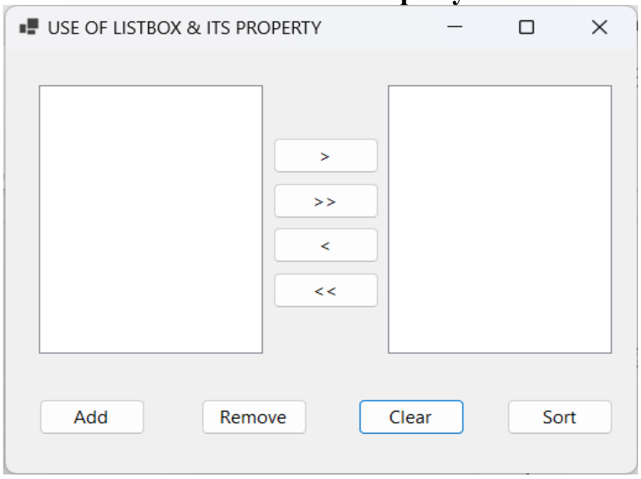
```
private void button7_Click(object sender, EventArgs e){
   listView1.Clear();
   listView2.Clear();
}

private void button8_Click(object sender, EventArgs e){
   if (sortFlag == true){
      listView1.Sorting = SortOrder.Ascending;
      listView2.Sorting = SortOrder.Ascending;
      sortFlag = false;
   }
   else{
      listView1.Sorting = SortOrder.Descending;
      listView2.Sorting = SortOrder.Descending;
      sortFlag = true;
   }
   listView1.Sort();
   listView2.Sort();
}
private void button1_Click(object sender, EventArgs e){
   // Check if there is any selected item
   if (listView1.SelectedItems.Count > 0){
      foreach (ListViewItem item in listView1.SelectedItems) {
         SelectedItem = item.Text; // Get the text of the item
      }
      listView2.Items.Add(SelectedItem);
      // Remove the selected item
      listView1.Items.Remove(listView1.SelectedItems[0]);
   }
   else {
      MessageBox.Show("No item selected to transfer.");
   }
}
private void button2_Click(object sender, EventArgs e) {
   foreach (ListViewItem item in listView1.Items) {
      ListViewItem list = (ListViewItem)item.Clone();
      listView2.Items.Add(list);
   }
   listView1.Clear();
}
private void button3_Click(object sender, EventArgs e){
   // Check if there is any selected item
   if (listView2.SelectedItems.Count > 0) {
      foreach (ListViewItem item in listView2.SelectedItems){
         SelectedItem = item.Text; // Get the text of the item
      }
      listView1.Items.Add(SelectedItem);
      // Remove the selected item
      listView2.Items.Remove(listView2.SelectedItems[0]);
   }
   else{
      MessageBox.Show("No item selected to transfer.");
   }
}
private void button4_Click(object sender, EventArgs e){
   foreach (ListViewItem item in listView2.Items){
      ListViewItem list = (ListViewItem)item.Clone();
      listView1.Items.Add(list);
```

```
            }
            listView2.Clear();
        }
    }
}
```

Output:



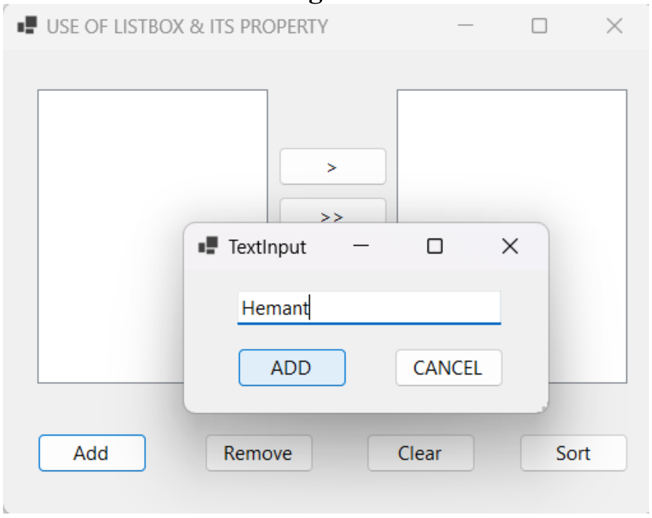
Listbox and Its Property :



Code for add Form

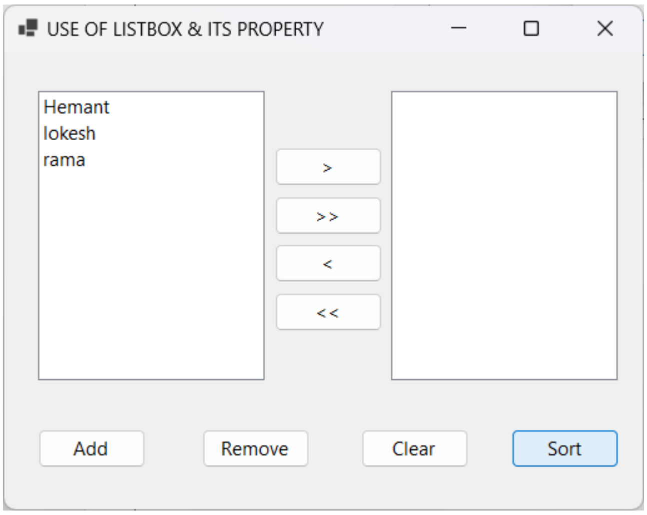
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace C_List2
{
    public partial class TextInput : Form{
        public string InputText { get; private set; }
        public TextInput(){
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e){
            // Set the InputText property to the text entered by the user
            InputText = textBox1.Text;
            this.DialogResult = DialogResult.OK;  // Close the dialog with OK result
            this.Close();  // Close the dialog
        }
        private void button2_Click(object sender, EventArgs e){
            // Close the dialog with Cancel result
            this.DialogResult = DialogResult.Cancel;
            this.Close();
        }
    }
}
```

Adding Items In List:



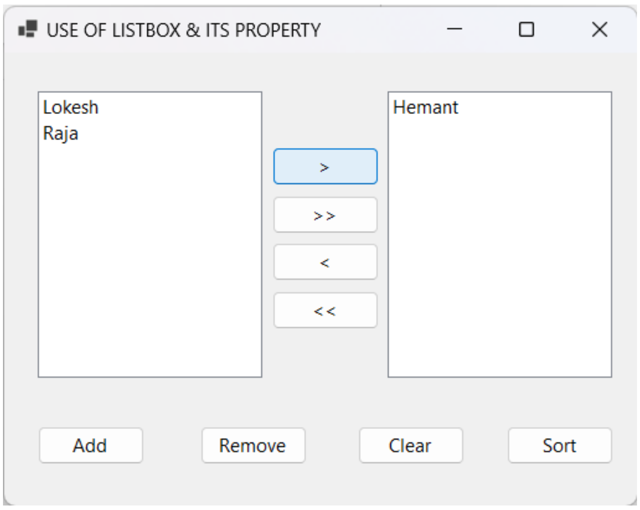
Hemant

Sort



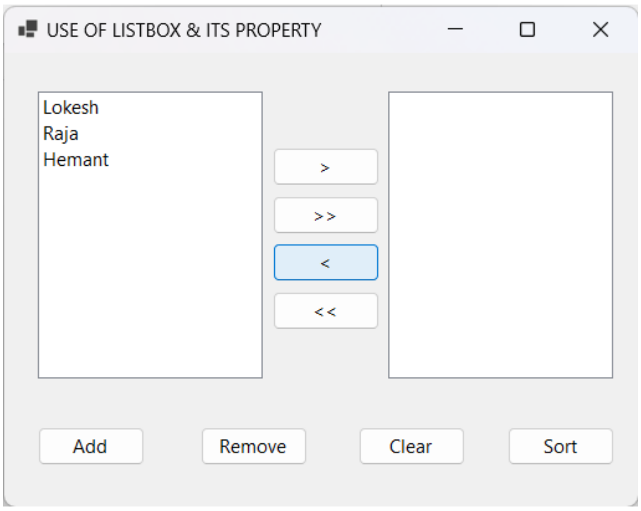
Hemant
lokesh
rama

Transfer Selected Item :



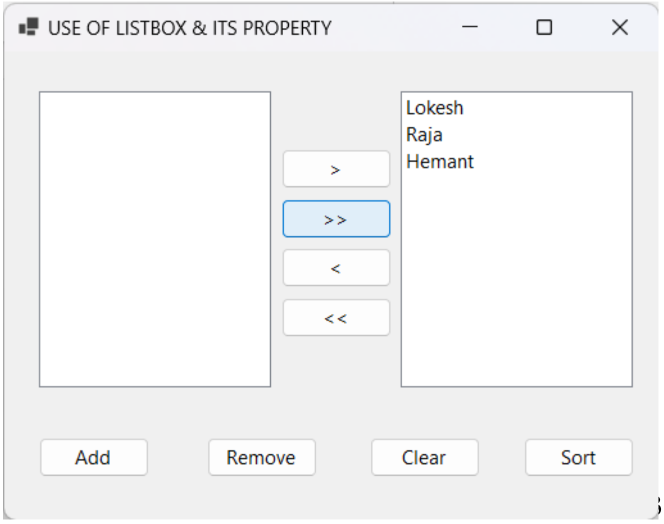
Lokesh
Raja

Hemant

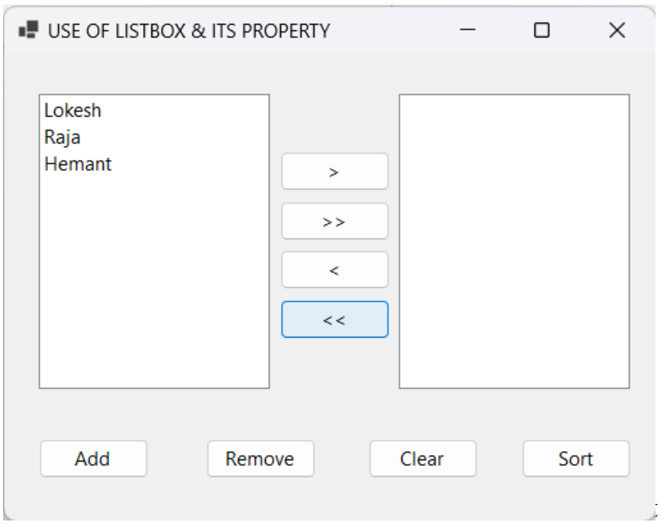


Lokesh
Raja
Hemant

Transfer ALL To Another List :

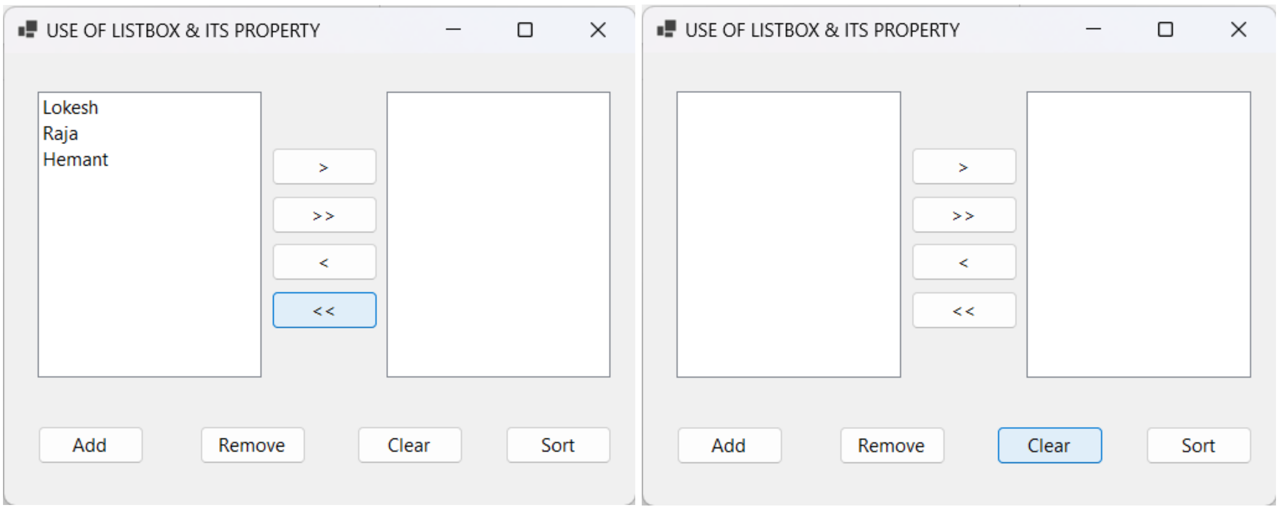


Lokesh
Raja
Hemant



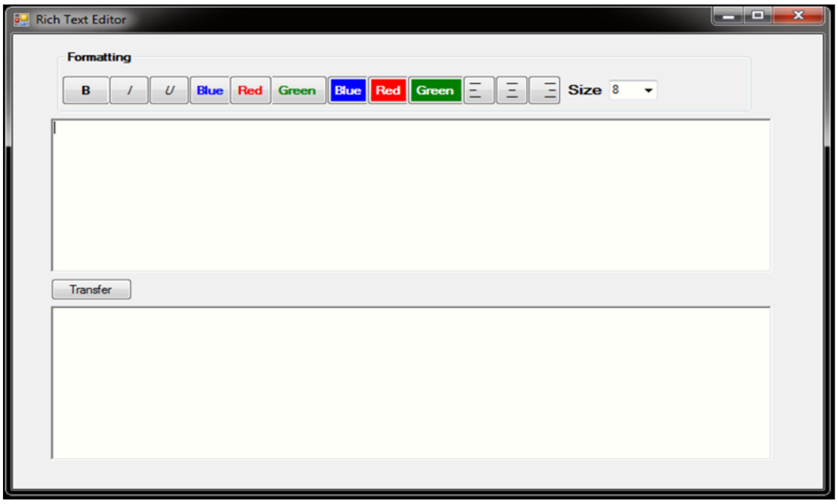
Lokesh
Raja
Hemant

Clear List :


USE OF LISTBOX & ITS PROPERTY
Lokesh
Raja
Hemant
>
>>
<
<<
Add
Remove
Clear
Sort
USE OF LISTBOX & ITS PROPERTY
>
>>
<
<<
Add
Remove
Clear
Sort

Practical No. 18

Design one form to create application like Rich text document using 1 Rich Text box and different buttons. When user presses any of this command buttons then the selected content of Rich textbox will be changed accordingly.

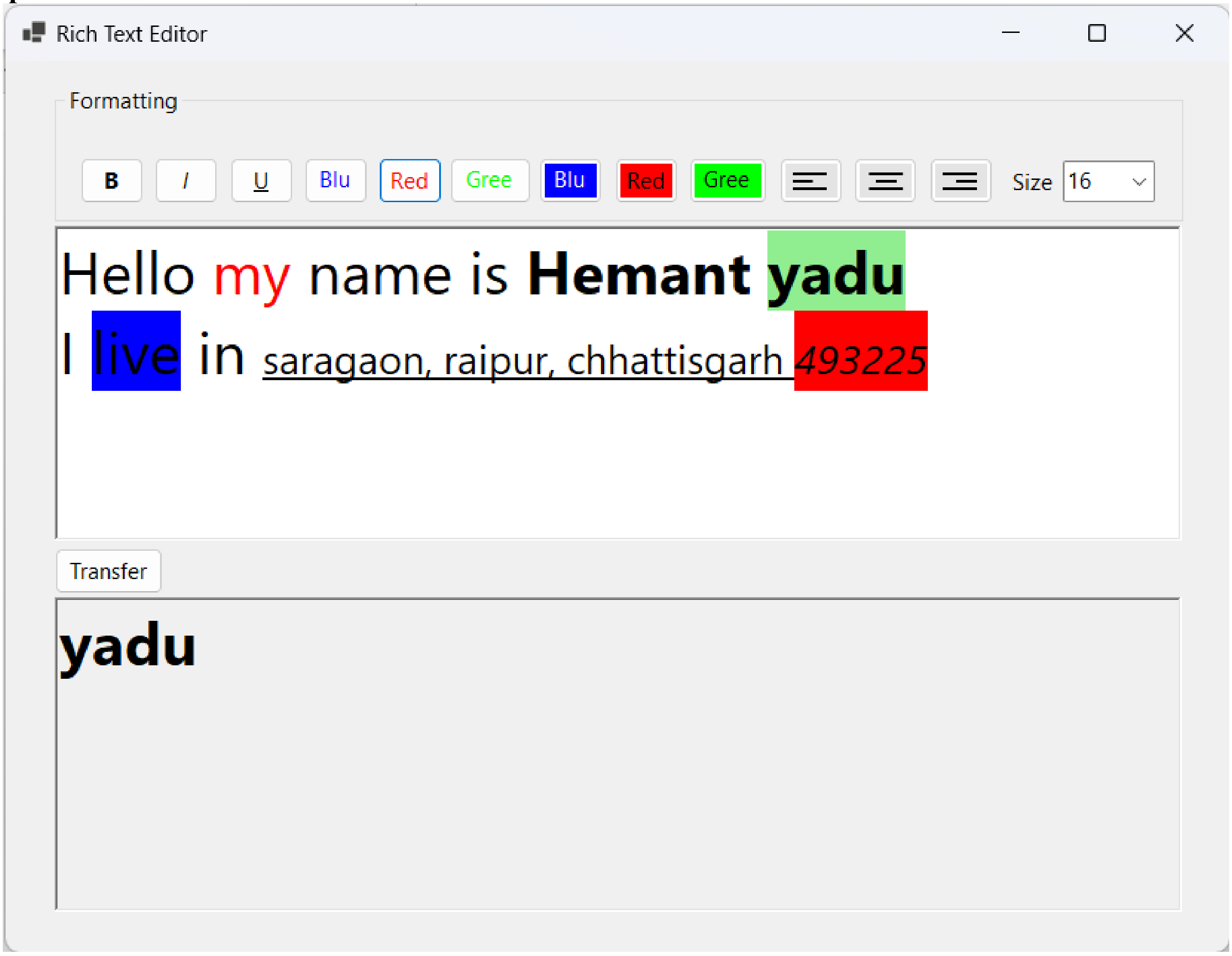


Code:

```
namespace C_List2
{
    public partial class Form5 : Form{
        public Form5(){
            InitializeComponent();
        }
        private void Form5_Load(object sender, EventArgs e){
            for (int i = 8; i < 40; i++){   //loop to add items in size.
                if (i % 2 == 0){
                    comboBox1.Items.Add(i);
                }
            }
        }
        private void button1_Click(object sender, EventArgs e){
            if (richTextBox1.SelectionLength > 0){     // Check if any text is selected
                // Get the current font of the selected text
                Font currentFont = richTextBox1.SelectionFont;
                if (currentFont != null){
                    // Toggle bold: if the text is bold, remove bold; otherwise, make it bold
                    FontStyle newFontStyle = currentFont.Bold ? FontStyle.Regular : FontStyle.Bold;
                    // Apply the new font style to the selected text
                    richTextBox1.SelectionFont = new Font(currentFont, newFontStyle);
                }
            }
            else{
                MessageBox.Show("Please select text to make bold.");
            }
        }
        private void button13_Click(object sender, EventArgs e){
            // Check if there is any text selected in richTextBox1
            if (!string.IsNullOrEmpty(richTextBox1.SelectedText)){
                // Check if the existing text in richTextBox2 ends with a space
                if (!richTextBox2.Text.EndsWith(" ") && richTextBox2.Text.Length > 0){
                    // If not, add a space before appending the new text
                    richTextBox2.AppendText(" ");
                }
                // Append the selected text from richTextBox1 to richTextBox2
        richTextBox2.SelectedRtf = richTextBox1.SelectedRtf;
        //alignment copy
        richTextBox2.SelectionAlignment = richTextBox1.SelectionAlignment;
        // Remove the selected text from richTextBox1
        richTextBox1.SelectedText = string.Empty;
    }
    else{
        MessageBox.Show("Please select some text to transfer.");
    }
}
private void button2_Click(object sender, EventArgs e){
    if (richTextBox1.SelectionLength > 0){ // Check if any text is selected
        // Get the current font of the selected text
        Font currentFont = richTextBox1.SelectionFont;
        if (currentFont != null){
            // Toggle Italic: if the text is italic, remove italic; otherwise, make it italic
            FontStyle newFontStyle = currentFont.Italic ? FontStyle.Regular : FontStyle.Italic;
            // Apply the new font style to the selected text
            richTextBox1.SelectionFont = new Font(currentFont, newFontStyle);
        }
    }
    else{
        MessageBox.Show("Please select text to make bold.");
    }
}
private void button3_Click(object sender, EventArgs e){
    if (richTextBox1.SelectionLength > 0){ // Check if any text is selected
        // Get the current font of the selected text
        Font currentFont = richTextBox1.SelectionFont;
        if (currentFont != null){
            // Toggle Underline: if the text is underline, remove underline; otherwise, make it UnderLine
            FontStyle newFontStyle = currentFont.Underline ? FontStyle.Regular : FontStyle.Underline;
            // Apply the new font style to the selected text
            richTextBox1.SelectionFont = new Font(currentFont, newFontStyle);
        }
    }
    else{
        MessageBox.Show("Please select text to make underline.");
    }
}
private void button4_Click(object sender, EventArgs e){
    if (richTextBox1.SelectionLength > 0){ // Check if any text is selected
        richTextBox1.SelectionColor = (richTextBox1.SelectionColor == Color.Blue) ? Color.Black : Color.Blue;
    }
    else{
        MessageBox.Show("Please select text to make color.");
    }
}
private void button5_Click(object sender, EventArgs e){
    if (richTextBox1.SelectionLength > 0){ // Check if any text is selected
        richTextBox1.SelectionColor = (richTextBox1.SelectionColor == Color.Red) ? Color.Black : Color.Red;
    }
    else{
        MessageBox.Show("Please select text to make color.");
    }
}
private void button6_Click(object sender, EventArgs e){
    if (richTextBox1.SelectionLength > 0){
        richTextBox1.SelectionColor = (richTextBox1.SelectionColor == Color.Green) ? Color.Black : Color.Green;
```

```
            }
            else{
                MessageBox.Show("Please select text to make color.");
            }
        }
        private void button7_Click(object sender, EventArgs e){
            if (richTextBox1.SelectionLength > 0){
                richTextBox1.SelectionBackColor = (richTextBox1.SelectionBackColor == Color.Blue) ? Color.Black :
Color.Blue;
            }
            else{
                MessageBox.Show("Please select text to make bg color.");
            }
        }
        private void button8_Click(object sender, EventArgs e)
        {
            if (richTextBox1.SelectionLength > 0){
                richTextBox1.SelectionBackColor = (richTextBox1.SelectionBackColor == Color.Red) ? Color.Black :
Color.Red;
            }
            else{
                MessageBox.Show("Please select text to make bg color.");
            }
        }
        private void button9_Click(object sender, EventArgs e){
            if (richTextBox1.SelectionLength > 0){
                richTextBox1.SelectionBackColor = (richTextBox1.SelectionBackColor == Color.Green) ? Color.Black :
Color.Green;
            }
            else{
                MessageBox.Show("Please select text to make bg color.");
            }
        }
        private void button10_Click(object sender, EventArgs e){
                richTextBox1.SelectionAlignment = HorizontalAlignment.Left;
        }
        private void button11_Click(object sender, EventArgs e)
        {
            richTextBox1.SelectionAlignment = HorizontalAlignment.Center;
        }
        private void button12_Click(object sender, EventArgs e){
            richTextBox1.SelectionAlignment = HorizontalAlignment.Right;
         }
        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e){
            float newFontSize = Convert.ToSingle(comboBox1.SelectedItem);
            Font currentFont = richTextBox1.SelectionFont;
            if (currentFont != null){
                // Apply the selected font size to the selected text
                richTextBox1.SelectionFont = new Font(currentFont.FontFamily, newFontSize, currentFont.Style);
            }
        }
    }
}
```

Output:



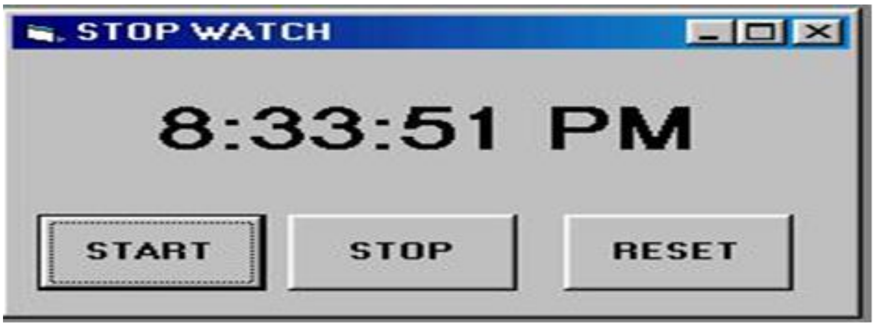
Hello my name is **Hemant** **yadu**

I live in saragaon, raipur, chhattisgarh *493225*

yadu

Practical No. 19

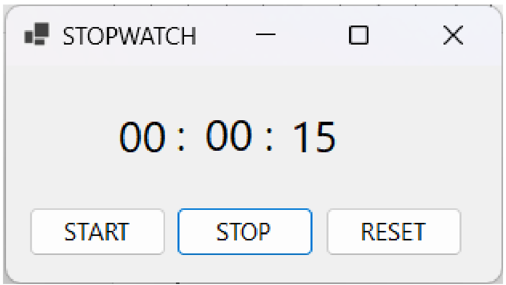
Design the digital watch using Timer Control.



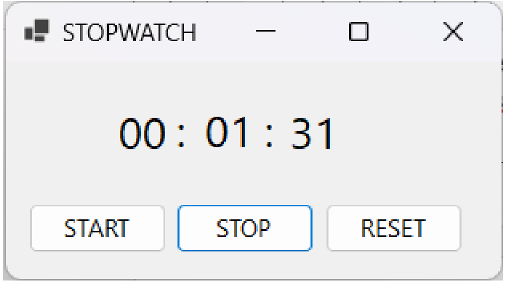
Code:

```
namespace C_List2
{
    public partial class Form6 : Form{
        // Counters for hours, minutes, and seconds
        private int hours = 0;
        private int minutes = 0;
        private int seconds = 0;
        public Form6(){
            InitializeComponent();
            timer1.Interval = 1000; // Set timer interval to 1000 ms (1 second)
            timer1.Tick += Timer1_Tick; // Attach Tick event handler
        }
        private void button1_Click(object sender, EventArgs e){
            // Start or resume the timer
            timer1.Start();
        }
        private void button2_Click(object sender, EventArgs e){
            // Stop (pause) the timer
            timer1.Stop();
        }
        private void button3_Click(object sender, EventArgs e){
            // Reset the timer to 00:00:00 and stop it
            timer1.Stop();
            hours = 0;
            minutes = 0;
            seconds = 0;
            UpdateTimeDisplay();
        }
        private void Timer1_Tick(object sender, EventArgs e){
            seconds++; // Increment seconds
            // Update minutes and hours based on seconds
            if (seconds == 60){
                seconds = 0;
                minutes++;
                if (minutes == 60){
                    minutes = 0;
                    hours++;
                }
            }
            // Update the display after each tick
            UpdateTimeDisplay();
        }
        private void UpdateTimeDisplay(){
            // Format and display time as hh:mm:ss
            hrs.Text = hours.ToString("D2") + ":";
            minute.Text = minutes.ToString("D2") + ":";
            second.Text = seconds.ToString("D2");
        }
    }
}
```

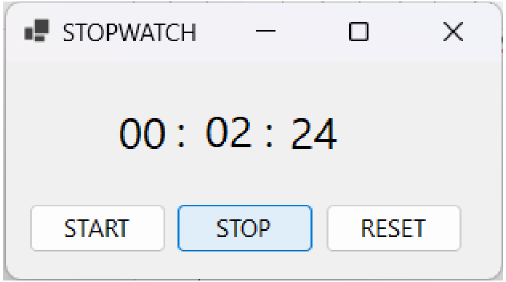
Output:



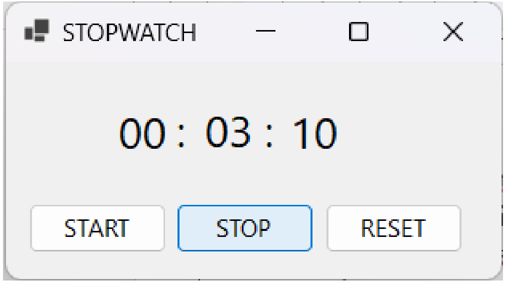
```
00 : 00 : 15
```



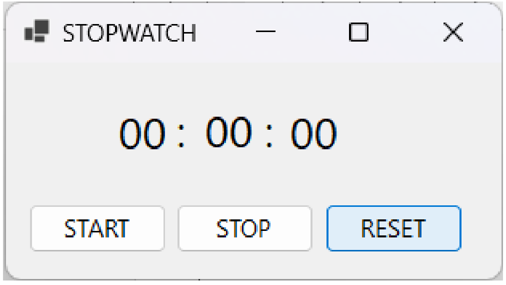
```
00 : 01 : 31
```



```
00 : 02 : 24
```



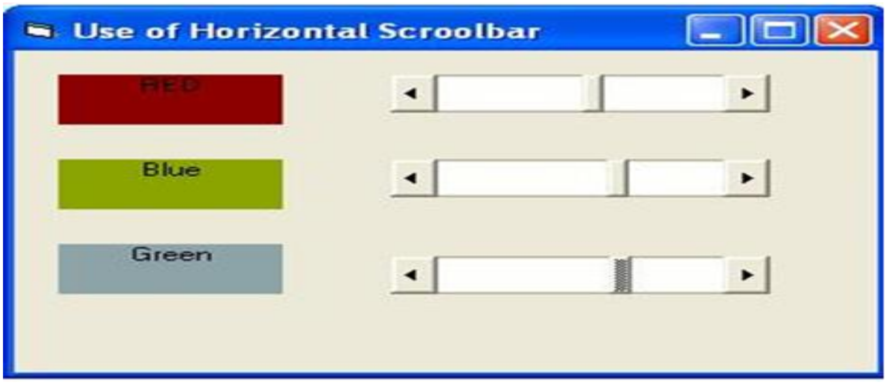
```
00 : 03 : 10
```



```
00 : 00 : 00
```

Practical No. 20

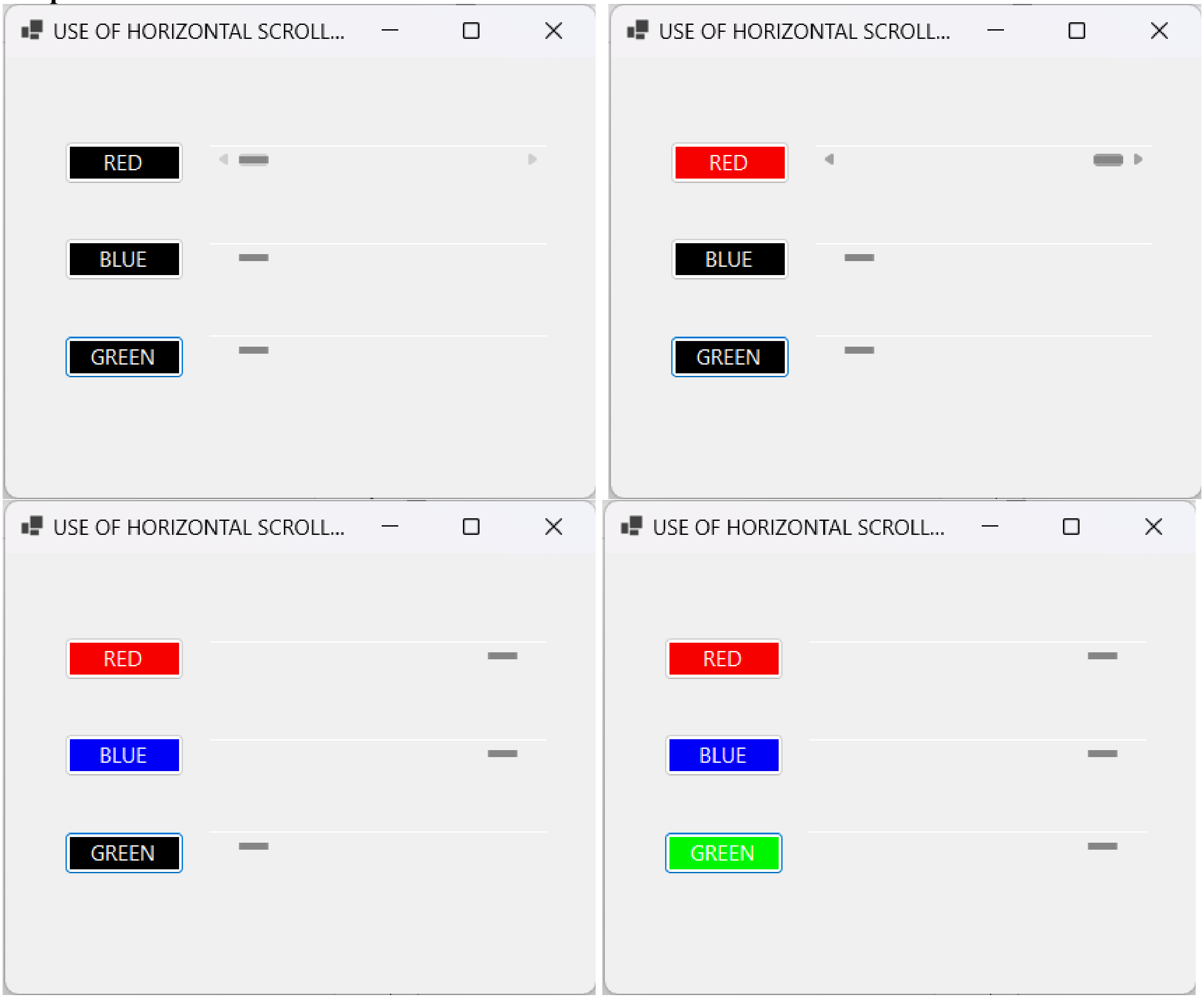
Design the following form using horizontal scrollbar. In this, when user click on particular scroll bar then back color of shape will be changed to Red, Green & Blue color.



Code:

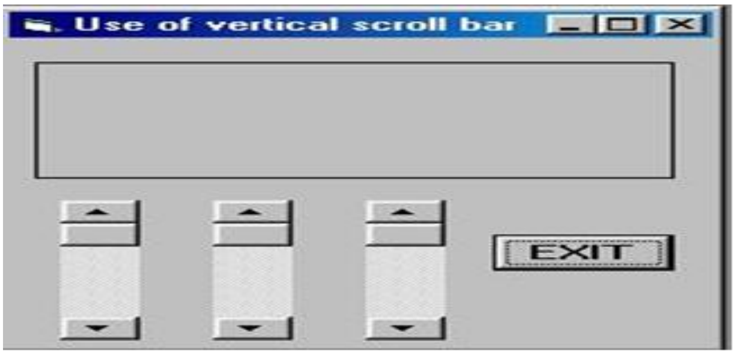
```
namespace C_List2{
   public partial class Form7 : Form{
      public Form7(){
         InitializeComponent();
      }
      private void Form7_Load(object sender, EventArgs e){
         button1.ForeColor = Color.White;
         button2.ForeColor = Color.White;
         button3.ForeColor = Color.White;
         // Set scrollbar properties (0 to 255 for color intensity)
         hScrollBar1.Minimum = 0;
         hScrollBar1.Maximum = 255;
         hScrollBar2.Minimum = 0;
         hScrollBar2.Maximum = 255;
         hScrollBar3.Minimum = 0;
         hScrollBar3.Maximum = 255;
         // Initialize button colors to dark
         button1.BackColor = Color.FromArgb(0, 0, 0); // Dark Red
         button2.BackColor = Color.FromArgb(0, 0, 0); // Dark Green
         button3.BackColor = Color.FromArgb(0, 0, 0); // Dark Blue
      }
      private void hScrollBar1_Scroll(object sender, ScrollEventArgs e){
         // Adjust the red component for button1 based on scrollbar value
         int redIntensity = hScrollBar1.Value;
         button1.BackColor = Color.FromArgb(redIntensity, 0, 0);
      }
      private void hScrollBar2_Scroll(object sender, ScrollEventArgs e){
         // Adjust the blue component for button3 based on scrollbar value
         int blueIntensity = hScrollBar2.Value;
         button2.BackColor = Color.FromArgb(0, 0, blueIntensity);
      }
      private void hScrollBar3_Scroll(object sender, ScrollEventArgs e){
         // Adjust the green component for button2 based on scrollbar value
         int greenIntensity = hScrollBar3.Value;
         button3.BackColor = Color.FromArgb(0, greenIntensity, 0);
      }
   }
}
```

Output:


USE OF HORIZONTAL SCROLL...
RED
BLUE
GREEN
USE OF HORIZONTAL SCROLL...
RED
BLUE
GREEN
USE OF HORIZONTAL SCROLL...
RED
BLUE
GREEN
USE OF HORIZONTAL SCROLL...
RED
BLUE
GREEN

Practical No. 21

Design the following form using vertical scrollbar. In this, when user click on particular scroll bar then back color of shape will be changed to Red, Green & Blue color.

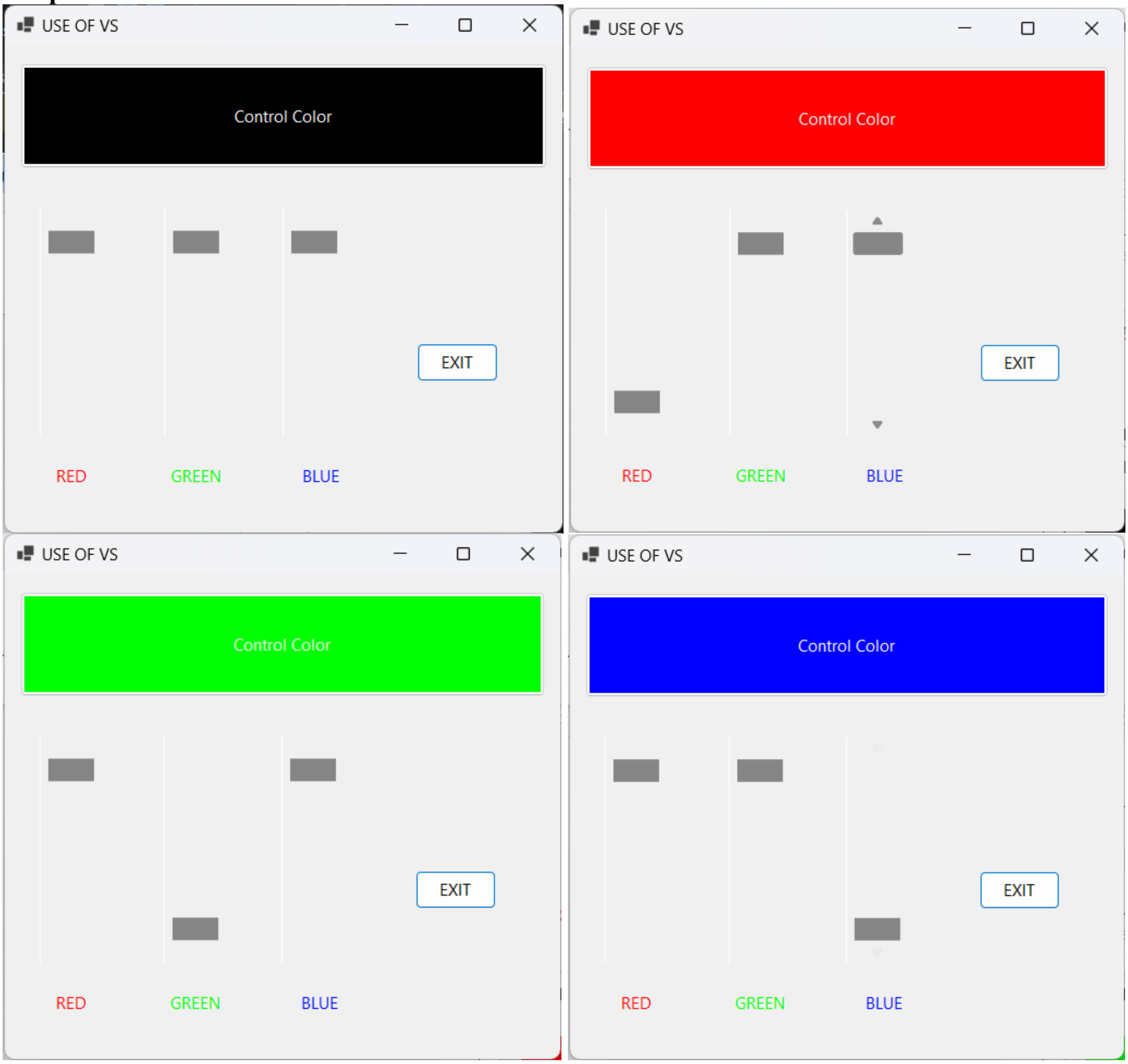


Code:

```
namespace C_List2{
   public partial class Form8 : Form{
      public Form8(){
         InitializeComponent();
         vScrollBar1.Scroll += vScrollBar1_Scroll;
         vScrollBar2.Scroll += vScrollBar2_Scroll;
         vScrollBar3.Scroll += vScrollBar3_Scroll;
      }
      private void Form8_Load(object sender, EventArgs e){
         // Set scrollbar properties to control RGB (0 to 255 for color intensity)
         vScrollBar1.Minimum = 0;
         vScrollBar1.Maximum = 255 + vScrollBar1.LargeChange - 1;
         vScrollBar2.Minimum = 0;
         vScrollBar2.Maximum = 255 + vScrollBar2.LargeChange - 1;
         vScrollBar3.Minimum = 0;
         vScrollBar3.Maximum = 255 + vScrollBar3.LargeChange - 1;
         // Set initial color of the button
         UpdateButtonColor();
      }
      private void vScrollBar1_Scroll(object sender, ScrollEventArgs e){
         // Update button color when red scrollbar changes
         UpdateButtonColor();
      }
      private void vScrollBar2_Scroll(object sender, ScrollEventArgs e){
         // Update button color when green scrollbar changes
         UpdateButtonColor();
      }
      private void vScrollBar3_Scroll(object sender, ScrollEventArgs e){
         // Update button color when blue scrollbar changes
         UpdateButtonColor();
      }
      private void UpdateButtonColor(){
         // Get the values from each scrollbar
         int red = vScrollBar1.Value;
         int green = vScrollBar2.Value;
         int blue = vScrollBar3.Value;
         // Set the button's background color using the RGB values
         button1.BackColor = Color.FromArgb(red, green, blue);
      }
```

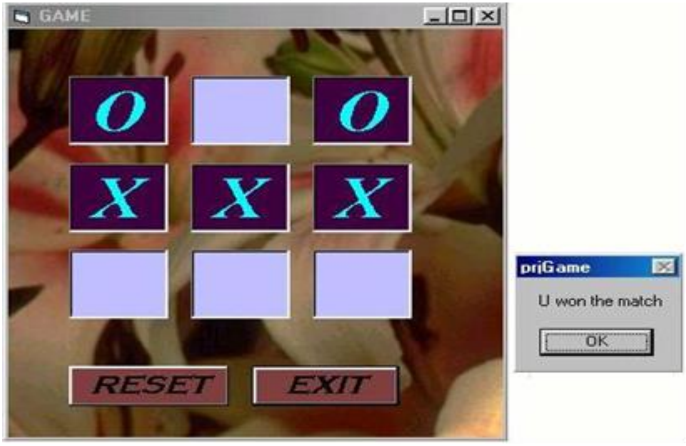
```
        private void button2_Click(object sender, EventArgs e){
           this.Close();
        }
    }
}
```

Output:



Practical No. 22

Design the following Tic-tac-toe game.

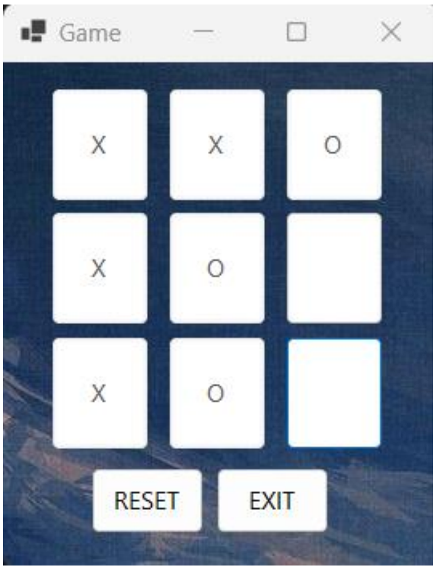


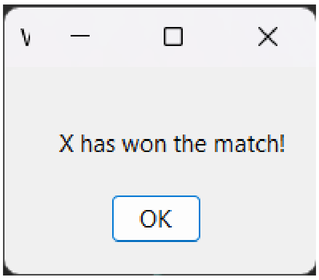
U won the match

Code:

```
namespace C_List2{
   public partial class Form9 : Form{
      private string currentPlayer = "X"; // Start with player X
      private Button[] buttons;
      public Form9(){
         InitializeComponent();
         InitializeTicTacToe();
      }
      private void InitializeTicTacToe(){
         buttons = new Button[] { button1, button2, button3, button4, button5, button6, button7, button8, button9 };
         foreach (var button in buttons){
            button.Text = "";
            button.Enabled = true;
            button.Click += Button_Click;  // Attach the click event handler
         }
      }
      private void Button_Click(object sender, EventArgs e){
         Button clickedButton = sender as Button;
         if (clickedButton != null && clickedButton.Text == ""){
            clickedButton.Text = currentPlayer;
            clickedButton.Enabled = false; // Disable button after clicked
            if (CheckForWinner()){
               ShowWinnerDialog(currentPlayer); // Show winner dialog
               ResetGame();
               return;
            }
            currentPlayer = currentPlayer == "X" ? "O" : "X"; // Switch players
         }
      }
      private bool CheckForWinner(){
         int[,] winningCombinations = new int[,]{
            { 0, 1, 2 }, { 3, 4, 5 }, { 6, 7, 8 }, // Rows
            { 0, 3, 6 }, { 1, 4, 7 }, { 2, 5, 8 }, // Columns
            { 0, 4, 8 }, { 2, 4, 6 }             // Diagonals
         };
         for (int i = 0; i < winningCombinations.GetLength(0); i++){
            if (buttons[winningCombinations[i, 0]].Text == currentPlayer &&
               buttons[winningCombinations[i, 1]].Text == currentPlayer &&
               buttons[winningCombinations[i, 2]].Text == currentPlayer){
                return true;
              }
           }
           return false;
        }
        private void ResetGame(){
           foreach (var button in buttons){
              button.Text = "";
              button.Enabled = true;
           }
           currentPlayer = "X"; // Start with player X again
        }
        private void ShowWinnerDialog(string winner){        // Method to show the winner dialog
           Winner winnerForm = new Winner(winner);
           winnerForm.ShowDialog(); // Show as a modal dialog
        }
        private void buttonReset_Click(object sender, EventArgs e){        // Reset Button Click event handler
           ResetGame(); // Reset the game board
        }
        private void buttonExit_Click(object sender, EventArgs e){        // Exit Button Click event handler
           this.Close(); // Close the current form (Form9)
        }
     }
  }
```

Output:





X has won the match!

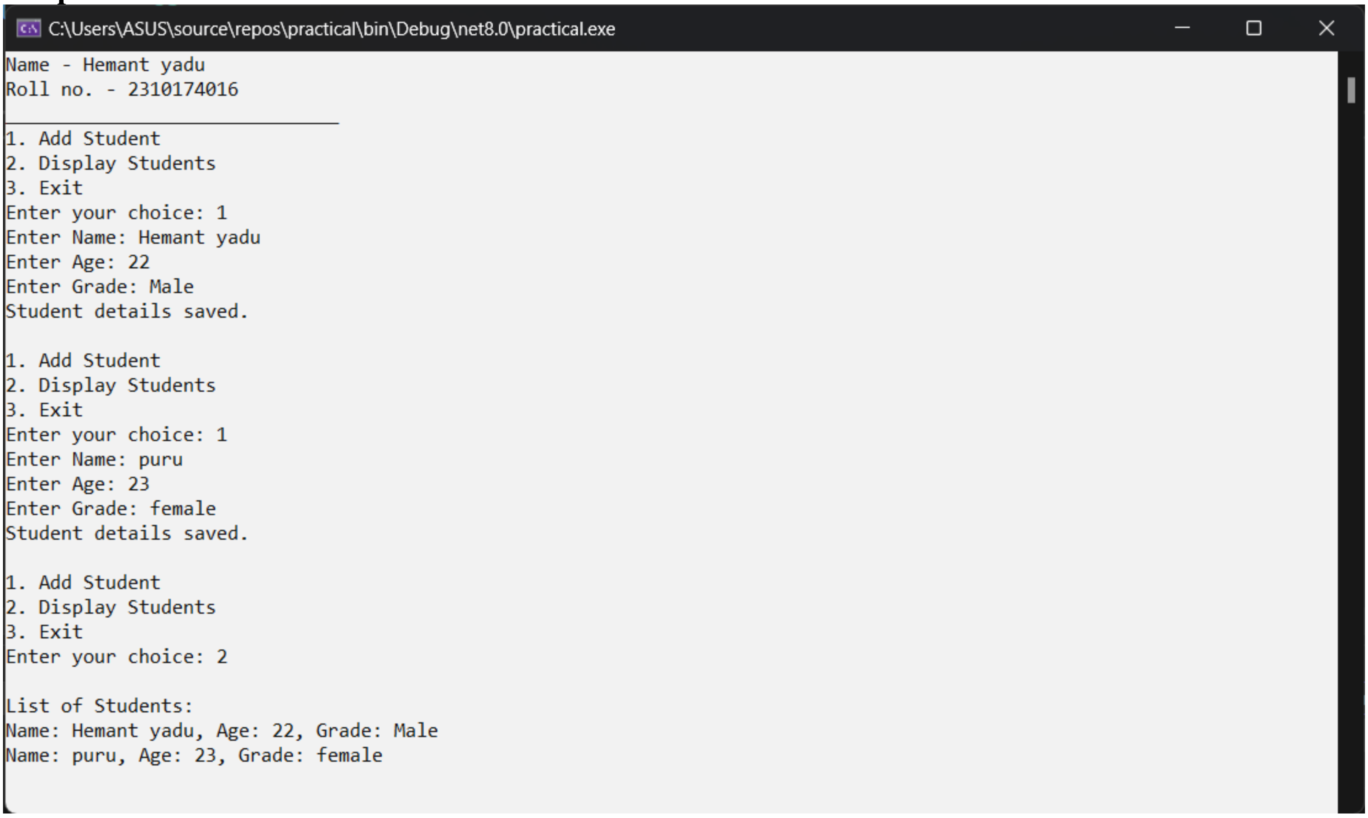
Practical No. 23

Write a .Net console application to store student details in file and display it in console.

Code:

```
using System;
using System.IO;
namespace practical{
   class Prog24{
      private string filePath = "students.txt";
      public Prog24(){
         int choice;
         do{
            Console.WriteLine("1. Add Student\n2. Display Students\n3. Exit");
            Console.Write("Enter your choice: ");
            choice = int.Parse(Console.ReadLine());
            if (choice == 1) AddStudent();
            else if (choice == 2) DisplayStudents();
            else if (choice != 3) Console.WriteLine("Invalid choice. Try again.");
         } while (choice != 3);
      }
      private void AddStudent(){
         Console.Write("Enter Name: ");
         string name = Console.ReadLine();
         Console.Write("Enter Age: ");
         int age = int.Parse(Console.ReadLine());
         Console.Write("Enter Grade: ");
         string grade = Console.ReadLine();
         File.AppendAllText(filePath, $"{name},{age},{grade}\n");
         Console.WriteLine("Student details saved.\n");
      }
      private void DisplayStudents(){
         if (File.Exists(filePath)){
            Console.WriteLine("\nList of Students:");
            foreach (string student in File.ReadAllLines(filePath)){
               var details = student.Split(',');
               Console.WriteLine($"Name: {details[0]}, Age: {details[1]}, Grade: {details[2]}");
            }
            Console.WriteLine();
         }
         else Console.WriteLine("No records found.\n");
      }
   }
}
```

Output:



C:\Users\ASUS\source\repos\practical\bin\Debug\net8.0\practical.exe

```
Name - Hemant yadu
Roll no. - 2310174016
______________________________
1. Add Student
2. Display Students
3. Exit
Enter your choice: 1
Enter Name: Hemant yadu
Enter Age: 22
Enter Grade: Male
Student details saved.

1. Add Student
2. Display Students
3. Exit
Enter your choice: 1
Enter Name: puru
Enter Age: 23
Enter Grade: female
Student details saved.

1. Add Student
2. Display Students
3. Exit
Enter your choice: 2

List of Students:
Name: Hemant yadu, Age: 22, Grade: Male
Name: puru, Age: 23, Grade: female
```

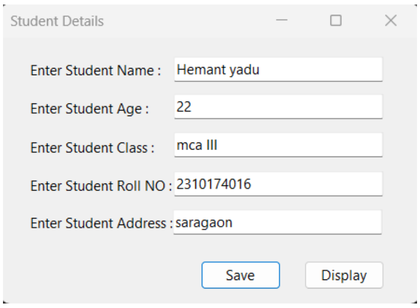
Practical No. 23

Write a .Net window application to enter the student details in form with proper GUI component and store in a file and display in another form.

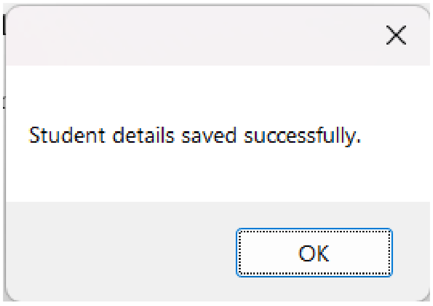
Code:

```
namespace C_List2
{
   public partial class Form10 : Form{
      public Form10(){
         InitializeComponent();
         button2.Click += new EventHandler(button2_Click);
      }
      private void button1_Click(object sender, EventArgs e){
         string name = textBox1.Text;
         string age = textBox2.Text;
         string studentClass = textBox3.Text;
         string rollNumber = textBox4.Text;
         string address = textBox5.Text;
         // Saving to file
         using (StreamWriter writer = new StreamWriter("student.txt", true)){
            writer.WriteLine($"{name},{age},{studentClass},{rollNumber},{address}");
         }

         MessageBox.Show("Student details saved successfully.");
      }
      private void button2_Click(object sender, EventArgs e) {
         Form11 form11 = new Form11();
         form11.ShowDialog();
      }
   }
}
```

Output:

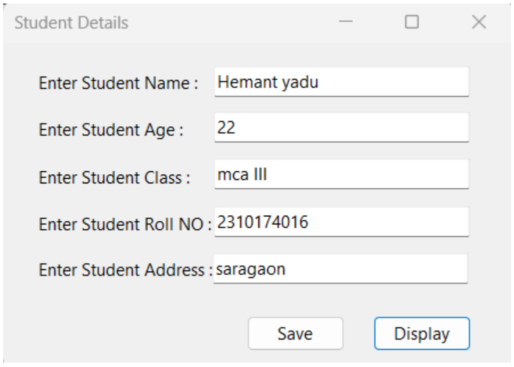


Enter Student Name : Hemant yadu
Enter Student Age : 22
Enter Student Class : mca III
Enter Student Roll NO : 2310174016
Enter Student Address : saragaon

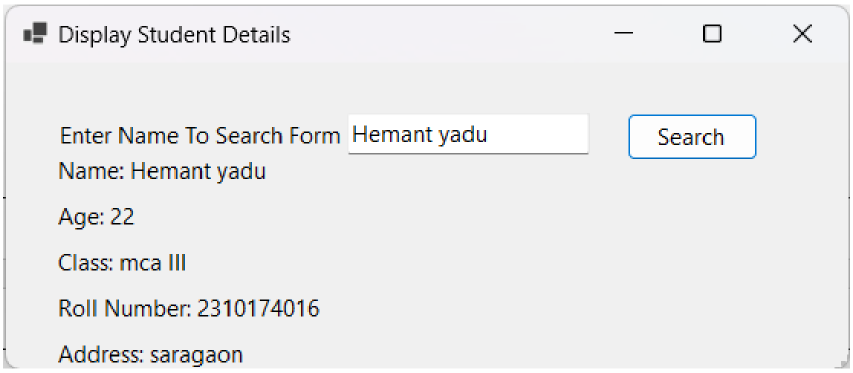


Student details saved successfully.

Display



Enter Student Name : Hemant yadu
Enter Student Age : 22
Enter Student Class : mca III
Enter Student Roll NO : 2310174016
Enter Student Address : saragaon



Enter Name To Search Form Hemant yadu
Name: Hemant yadu
Age: 22
Class: mca III
Roll Number: 2310174016
Address: saragaon