

MANGALORE



UNIVERSITY

MASTER OF COMPUTER APPLICATION

MCAP108: .NET TECHNOLOGY LAB

SUBMITTED

BY

I SEMESTER MCA Students

SUBMITTED

TO

Ms. Priyanka K.

Department of Computer Science

Lecturers, In-charge:

1.

2.

Mangalore University
Dept. of Post-Graduate Studies and Research in Computer Science
Mangalagangothri - 574199

	INDEX	
Sl No	Program List	Page No
CONSOLE APPLICATIONS		
1	C# Program to Print a Binary Triangle.	1
2	C# Program to Check Whether the Entered Number is an Amicable Number or Not.	2
3	C# Program to Illustrate Multilevel Inheritance with Virtual Methods (displaying student details).	4
4	C# Program to Create a Gray Code.	8
5	C# program to calculate volume of 2 boxes and find the resultant volume after addition of 2 boxes by implementing operator overloading.	9
6	C# Program to Implement Principles of Delegates (converting input string to uppercase first, last and entire string).	11
7	C# Program to Generate Register Number automatically for 100 Students using Static Constructor.	13
8	C# Program to Find the Frequency of the Word "is" in a given Sentence.	15
9	C# program that benchmarks 2D, jagged array allocation.	17
10	C# Program to Find the Sum of the Values on Diagonal of the Matrix.	19
11	C# Program to Create a File, Check the Existence of a File and Read the Contents of the File.	21
12	C# Program to Perform File Comparison.	25
13	C# Program to Implement IComparable Interface.	27
14	C# Program to Create Thread Pools.	29
15	Write a C# program to demonstrate error handling using Try, Catch and Finally block.	31

	INDEX	
Sl No	Program List	Page No
WINDOWS APPLICATIONS		
16	C# Program to Convert Digits to Words.	33
17	C# Program to Implement PhoneBook.	36
18	C# Program to Perform Reversal, Padding and Trimming Operations on string.	40
19	C# Program to Create a Progress Bar Control.	42
20	C# Program to perform a number guessing game.	44
21	Develop a WinForm application to create flat clock.	46
22	Develop an application to create a notepad.	50
23	Develop an application to construct a graphical binary tree where you need to create, add, search and remove nodes.	53

1. C# Program to Print a Binary Triangle.

using System;

namespace Exercises

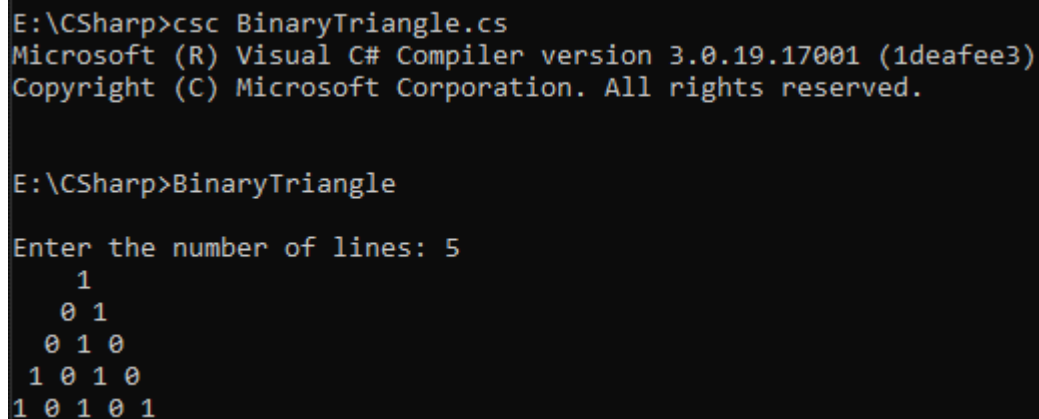
```
{
    class BinaryTriangle
    {
        static void Main(string[] args)
        {
            int number,digit=1;
            Console.Write("\nEnter the number of lines: ");
            number = Convert.ToInt32(Console.ReadLine());

            for(int i=1; i<=number; i++)
            {
                for(int space=number-i; space>0; space--)
                {
                    Console.Write(" ");
                }

                for(int j=0; j<i; j++)
                {
                    Console.Write(digit + " ");
                    digit = (digit==1) ? 0:1;
                }

                Console.Write("\n");
            }
        }
    }
}
```

OUTPUT



```
E:\CSharp>csc BinaryTriangle.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deaf3e3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>BinaryTriangle

Enter the number of lines: 5
  1
 0 1
0 1 0
1 0 1 0
1 0 1 0 1
```

2. C# Program to Check Whether the Entered Number is an Amicable Number or Not.

using System;

namespace Exercises

```
{
    class AmicableNumber
    {
        static void Main(string[] args)
        {
            int num1, num2, sum1=0, sum2=0;
            Console.WriteLine("\n-----AMICABLE NUMBERS-----\n");
            Console.Write("\nEnter the first number: ");
            num1 = Convert.ToInt32(Console.ReadLine());
            Console.Write("\nEnter the second number: ");
            num2 = Convert.ToInt32(Console.ReadLine());

            for(int i=1; i<num1; i++)
            {
                if(num1%i == 0)
                {
                    sum1 += i;
                }
            }

            for(int i=1; i<num2; i++)
            {
                if(num2%i == 0)
                {
                    sum2 += i;
                }
            }

            if(sum1 == num2 && sum2 == num1)
            {
                Console.WriteLine("\nThe numbers {0} and {1} are amicable.", num1, num2);
            }
            else
            {
                Console.WriteLine("\nThe numbers {0} and {1} are not amicable.", num1, num2);
            }
        }
    }
}
```

OUTPUT

```
E:\CSharp>csc AmicableNumber.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>AmicableNumber

-----AMICABLE NUMBERS-----

Enter the first number: 220
Enter the second number: 284
The numbers 220 and 284 are amicable.

E:\CSharp>AmicableNumber

-----AMICABLE NUMBERS-----

Enter the first number: 220
Enter the second number: 294
The numbers 220 and 294 are not amicable.
```

3. C# Program to Illustrate Multilevel Inheritance with Virtual Methods (displaying student details).

using System;

```
namespace Exercises
{
    class PersonalDetails
    {
        string name;
        int age;
        string gender;

        public PersonalDetails(string name, int age, string gender)
        {
            this.name = name;
            this.age = age;
            this.gender = gender;
        }

        public virtual void Display()
        {
            Console.WriteLine("\n----- PERSONAL DETAILS ----- \n");
            Console.WriteLine("Name           : " + name);
            Console.WriteLine("Age           : " + age);
            Console.WriteLine("Gender        : " + gender);
        }
    }

    class CourseDetails : PersonalDetails
    {
        int regNo;
        string course;
        int semester;

        public CourseDetails(string name, int age, string gender, int regNo, string course, int semester)
            : base(name, age, gender)
        {
            this.regNo = regNo;
            this.course = course;
            this.semester = semester;
        }

        public override void Display()
        {
            base.Display();
            Console.WriteLine("\n----- COURSE DETAILS ----- \n");
        }
    }
}
```

```

        Console.WriteLine("Register Number   : " + regNo);
        Console.WriteLine("Course           : " + course);
        Console.WriteLine("Semester         : " + semester);
    }
}

```

```

class MarksDetails : CourseDetails

```

```

{
    int[] marks = new int[5];
    int total;
    float average;
    string grade;
    int flagFail;

```

```

    public MarksDetails(string name, int age, string gender, int regNo, string course, int semester,
int[] marks) : base(name, age, gender, regNo, course, semester)

```

```

    {
        total = 0;
        for(int i=0; i<5; i++)
        {
            this.marks[i] = marks[i];
            total += marks[i];

```

```

            if(marks[i]<35)
            {
                flagFail = 1;
            }
        }
    }

```

```

        Calculate();
    }

```

```

private void Calculate()

```

```

{
    average = total/5;

    if(flagFail == 1 || average<40)
        grade = "Fail";
    else if(average>=70)
        grade = "Distinction";
    else if(average>=60)
        grade = "First Class";
    else if(average>=50)
        grade = "Second Class";
}

```



```

else
    grade = "Pass Class";
}

public override void Display()
{
    base.Display();
    Console.WriteLine("\n----- MARKS DETAILS -----\\n");
    Console.Write("Marks in 5 subjects: ");
    for(int i=0; i<5; i++)
        Console.Write(marks[i] + " ");
    Console.WriteLine();
    Console.WriteLine("Total          : " + total);
    Console.WriteLine("Average          : " + average);
    Console.WriteLine("Grade           : " + grade);
}
}

class MultiLevel
{
    public static void Main(string[] args)
    {
        MarksDetails Student1 = new MarksDetails("Abhijith", 22, "Male", 20190001, "MCA", 5,
new int[]{77,80,98,95,90});
        Student1.Display();
    }
}
}

```

OUTPUT

```
E:\CSharp>csc MultiLevel.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>MultiLevel

----- PERSONAL DETAILS -----
Name           : Abhijith
Age            : 22
Gender         : Male

----- COURSE DETAILS -----
Register Number : 20190001
Course          : MCA
Semester        : 5

----- MARKS DETAILS -----
Marks in 5 subjects: 77    80    98    95    90
Total            : 440
Average          : 88
Grade            : Distinction
```

4. C# Program to Create a Gray Code.

```
using System;
```

```
namespace Exercises
```

```
{
```

```
    class GrayCode
```

```
    {
```

```
        static int getGray(int n)
```

```
        {
```

```
            return n^(n>>1);
```

```
        }
```

```
        static void Main(string[] args)
```

```
        {
```

```
            int InputNum, GrayNum;
```

```
            Console.WriteLine("\nEnter the decimal number: ");
```

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

```
            Console.WriteLine("\nBinary equivalent of {0}: {1}", InputNum, Convert.ToString(InputNum, 2));
```

```
            GrayNum = getGray(InputNum);
```

```
            Console.WriteLine("\nGray Code equivalent of {0}: {1}", InputNum, Convert.ToString(GrayNum, 2));
```

```
        }
```

```
    }
```

```
}
```

OUTPUT

```
E:\CSharp>csc GrayCode.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>GrayCode

Enter the decimal number: 123

Binary equivalent of 123: 1111011

Gray Code equivalent of 123: 1000110
```

5. C# program to calculate volume of 2 boxes and find the resultant volume after addition of 2 boxes by implementing operator overloading.

```
using System;

namespace Exercises
{
    class Box
    {
        float width;
        float height;
        float length;

        public float Volume
        {
            get { return width * height * length; }
        }

        public Box(float width, float height, float length)
        {
            this.width = width;
            this.height = height;
            this.length = height;
        }

        public static float operator +(Box box1, Box box2)
        {
            return box1.Volume + box2.Volume;
        }

        public override String ToString()
        {
            return "box with width " + width + ", height " + height + " and length " + length;
        }
    }

    class OperatorOverloading
    {
        public static void Main()
        {
            Box box1 = new Box(10, 20, 30);
            Box box2 = new Box(25, 32, 15);

            Console.WriteLine("Volume of {0} is: {1}", box1, box1.Volume);
            Console.WriteLine("Volume of {0} is: {1}", box2, box2.Volume);
        }
    }
}
```

```
        Console.WriteLine("Volume after adding boxes: {0}", box1 + box2);  
    }  
}  
}
```

OUTPUT

```
E:\CSharp>csc AbstractProperties.cs  
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafef3)  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
E:\CSharp>AbstractProperties  
Area of Circle of 20 is 1256.64  
Area of Square of side 20 is 400.00  
Area of Rectangle of length 20 and width 30 is 600.00
```

6. C# Program to Implement Principles of Delegates (converting input string to uppercase first, last and entire string).

```
using System;

namespace Exercises
{
    class Delegates
    {
        delegate string UppercaseDelegate(string input);
        static string UppercaseFirst(string input)
        {
            char[] buffer = input.ToCharArray();
            buffer[0] = char.ToUpper(buffer[0]);
            return new string(buffer);
        }
        static string UppercaseLast(string input)
        {
            char[] buffer = input.ToCharArray();
            buffer[buffer.Length - 1] = char.ToUpper(buffer[buffer.Length - 1]);
            return new string(buffer);
        }
        static string UppercaseAll(string input)
        {
            return input.ToUpper();
        }
        static void WriteOutput(string input, UppercaseDelegate del)
        {
            Console.WriteLine("Input String: {0}", input);
            Console.WriteLine("Output String: {0}", del(input));
        }
        static void Main()
        {
            WriteOutput("tom ", new UppercaseDelegate(UppercaseFirst));
            WriteOutput("tom", new UppercaseDelegate(UppercaseLast));
            WriteOutput("tom", new UppercaseDelegate(UppercaseAll));
            Console.ReadLine();
        }
    }
}
```

OUTPUT

```
E:\CSharp>csc Delegates.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>Delegates
Input String: tom
Output String: Tom
Input String: tom
Output String: toM
Input String: tom
Output String: TOM
```

7. C# Program to Generate Register Number automatically for 100 Students using Static Constructor.

```
using System;

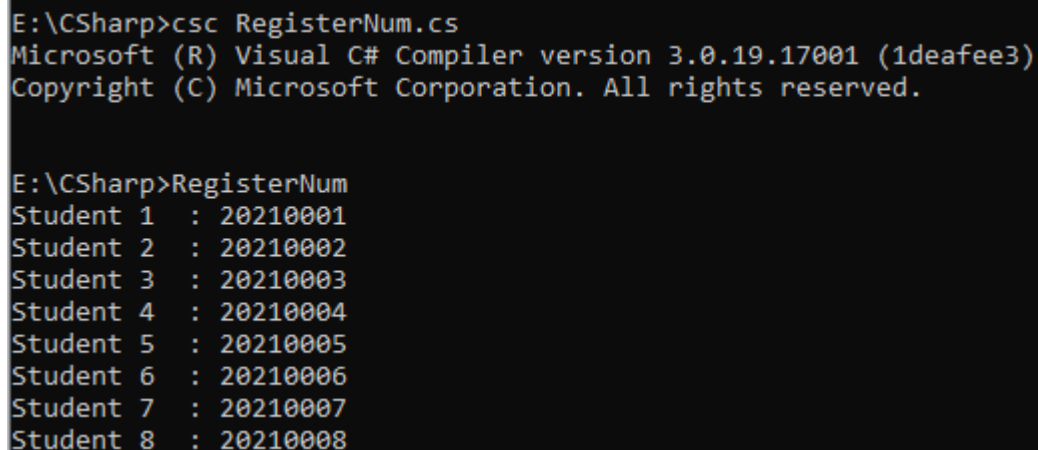
namespace Exercises
{
    class RegisterNum
    {
        int regNo;
        static int startNum;

        static RegisterNum()
        {
            startNum = 20210000;
        }

        RegisterNum()
        {
            regNo = ++startNum;
        }

        public static void Main(string[] args)
        {
            for(int i = 0; i<100; i++)
            {
                RegisterNum Student = new RegisterNum();
                Console.WriteLine("Student {0} : {1}", i+1, Student.regNo);
            }
        }
    }
}
```

OUTPUT



```
E:\CSharp>csc RegisterNum.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>RegisterNum
Student 1 : 20210001
Student 2 : 20210002
Student 3 : 20210003
Student 4 : 20210004
Student 5 : 20210005
Student 6 : 20210006
Student 7 : 20210007
Student 8 : 20210008
```


Student 9 : 20210009
Student 10 : 20210010
Student 11 : 20210011
Student 12 : 20210012
Student 13 : 20210013
Student 14 : 20210014
Student 15 : 20210015
Student 16 : 20210016
Student 17 : 20210017
Student 18 : 20210018
Student 19 : 20210019
Student 20 : 20210020
Student 21 : 20210021
Student 22 : 20210022
Student 23 : 20210023
Student 24 : 20210024
Student 25 : 20210025
Student 26 : 20210026
Student 27 : 20210027
Student 28 : 20210028
Student 29 : 20210029
Student 30 : 20210030
Student 31 : 20210031
Student 32 : 20210032
Student 33 : 20210033
Student 34 : 20210034
Student 35 : 20210035
Student 36 : 20210036
Student 37 : 20210037
Student 38 : 20210038
Student 39 : 20210039
Student 40 : 20210040
Student 41 : 20210041
Student 42 : 20210042
Student 43 : 20210043
Student 44 : 20210044
Student 45 : 20210045
Student 46 : 20210046
Student 47 : 20210047
Student 48 : 20210048
Student 49 : 20210049
Student 50 : 20210050
Student 51 : 20210051
Student 52 : 20210052
Student 53 : 20210053
Student 54 : 20210054

Student 55 : 20210055
Student 56 : 20210056
Student 57 : 20210057
Student 58 : 20210058
Student 59 : 20210059
Student 60 : 20210060
Student 61 : 20210061
Student 62 : 20210062
Student 63 : 20210063
Student 64 : 20210064
Student 65 : 20210065
Student 66 : 20210066
Student 67 : 20210067
Student 68 : 20210068
Student 69 : 20210069
Student 70 : 20210070
Student 71 : 20210071
Student 72 : 20210072
Student 73 : 20210073
Student 74 : 20210074
Student 75 : 20210075
Student 76 : 20210076
Student 77 : 20210077
Student 78 : 20210078
Student 79 : 20210079
Student 80 : 20210080
Student 81 : 20210081
Student 82 : 20210082
Student 83 : 20210083
Student 84 : 20210084
Student 85 : 20210085
Student 86 : 20210086
Student 87 : 20210087
Student 88 : 20210088
Student 89 : 20210089
Student 90 : 20210090
Student 91 : 20210091
Student 92 : 20210092
Student 93 : 20210093
Student 94 : 20210094
Student 95 : 20210095
Student 96 : 20210096
Student 97 : 20210097
Student 98 : 20210098
Student 99 : 20210099
Student 100 : 20210100

8. C# Program to Find the Frequency of the Word "is" in a given Sentence.

```
using System;
namespace Exercises
{
    class FrequencyIS
    {
        static void Main(string[] args)
        {
            int count=0;
            string inputString;
            Console.WriteLine("\n----- Frequency of word 'is' -----");
            Console.Write("\n Enter the input string: ");
            inputString = Console.ReadLine();
            char[] separator = { ',', ' ', '!', '!', '\n' };
            string testString = inputString.ToLower();
            String[] outcomes = testString.Split(separator);

            foreach(String s in outcomes)
            {
                Console.WriteLine(s);
                if(s == "is")
                    count++;
            }

            Console.WriteLine("\n Number of 'is' in '" + inputString + "' is: " + count);
        }
    }
}
```

OUTPUT

```
E:\CSharp>csc FrequencyIS.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafef3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>FrequencyIS

----- Frequency of word 'is' -----

Enter the input string: This statement is to check frequency of word is.
this
statement
is
to
check
frequency
of
word
is

Number of 'is' in 'This statement is to check frequency of word is.' is: 2
```

9. C# program that benchmarks 2D, jagged array allocation.

```
using System;
using System.Diagnostics;

namespace Exercises
{
    class BenchmarkAllocation
    {
        const int _max = 100000;
        static void Main(string[] args)
        {
            var Arr2D = new int[100,100];
            var ArrJagged = new int[100][];

            for(int i=0; i<100; i++)
            {
                ArrJagged[i] = new int[100];
            }

            var Stopwatch2D = Stopwatch.StartNew();
            for(int i=0; i<_max; i++)
            {
                for(int j=0; j<100; j++)
                {
                    for(int k=0; k<100; k++)
                    {
                        Arr2D[j, k] = k;
                    }
                }
            }
            Stopwatch2D.Stop();

            var StopwatchJagged = Stopwatch.StartNew();
            for(int i=0; i<_max; i++)
            {
                for(int j=0; j<100; j++)
                {
                    for(int k=0; k<100; k++)
                    {
                        ArrJagged[j][k] = k;
                    }
                }
            }
            StopwatchJagged.Stop();
        }
    }
}
```

```
Console.WriteLine("\n Time taken for allocation in case of 2D array: ");
Console.WriteLine(Stopwatch2D.Elapsed.TotalMilliseconds + " milliseconds");
Console.WriteLine("\n Time taken for allocation in case of Jagged array: ");
Console.WriteLine(StopwatchJagged.Elapsed.TotalMilliseconds + " milliseconds");
}
}
}
```

OUTPUT

```
E:\CSharp>csc BenchmarkAllocation.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>BenchmarkAllocation

Time taken for allocation in case of 2D array: 8243.7283 milliseconds

Time taken for allocation in case of Jagged array: 3764.0026 milliseconds
```

10. C# Program to Find the Sum of the Values on Diagonal of the Matrix.

using System;

namespace Exercises

{

class SumOfDiagonals

{

static void Main(string[] args)

{

int MaxRow, MaxCol, Sum=0;

int[,] Matrix;

Console.WriteLine("\n----- SUM OF DIAGONAL OF A MATRIX ----- \n");

Console.Write("\nEnter the number of rows: ");

MaxRow = Convert.ToInt32(Console.ReadLine());

Console.Write("\nEnter the number of columns: ");

MaxCol = Convert.ToInt32(Console.ReadLine());

if(MaxRow != MaxCol)

{

Console.WriteLine("\nThe Dimensions entered are not of Square Matrix.");

Console.WriteLine("\nExiting the Program..");

return;

}

Matrix = new int[MaxRow, MaxCol];

for(int i=0; i<MaxRow; i++)

{

for(int j=0; j<MaxCol; j++)

{

Console.Write("\nEnter the ({0},{1})th element of the matrix: ", (i+1), (j+1));

Matrix[i, j] = Convert.ToInt32(Console.ReadLine());

}

}

Console.WriteLine("\nThe entered Matrix is: ");

for(int i=0; i<MaxRow; i++)

{

for(int j=0; j<MaxCol; j++)

{

Console.Write(" " + Matrix[i, j]);

if(i == j)

{

```

        Sum += Matrix[i,j];
    }
}
Console.WriteLine();
}

Console.WriteLine("\nThe Sum of Diagonal is " + Sum);
}
}
}

```

OUTPUT

```

E:\CSharp>csc SumOfDiagonal.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>SumOfDiagonal

----- SUM OF DIAGONAL OF A MATRIX -----

Enter the number of rows: 3
Enter the number of columns: 3
Enter the (1,1)th element of the matrix: 1
Enter the (1,2)th element of the matrix: 2
Enter the (1,3)th element of the matrix: 3
Enter the (2,1)th element of the matrix: 4
Enter the (2,2)th element of the matrix: 5
Enter the (2,3)th element of the matrix: 6
Enter the (3,1)th element of the matrix: 7
Enter the (3,2)th element of the matrix: 8
Enter the (3,3)th element of the matrix: 9

The entered Matrix is:
  1  2  3
  4  5  6
  7  8  9

The Sum of Diagonal is 15

```

11. C# Program to Create a File, Check the Existence of a File and Read the Contents of the File.

```
using System;
using System.IO;

namespace Exercises
{
    class FileRead
    {
        public static void Main()
        {
            string fileName;
            while (true)
            {
                Console.WriteLine("\n ----- MENU ----- \n");
                Console.WriteLine("\n 1.Create a File ");
                Console.WriteLine("\n 2.Existence of the File ");
                Console.WriteLine("\n 3.Read the contents of the File ");
                Console.WriteLine("\n 4.Exit ");
                Console.Write("\n Enter your choice : ");
                int ch = int.Parse(Console.ReadLine());
                switch (ch)
                {
                    case 1:
                        Console.Write("\n Enter the file name to create: ");
                        fileName = Console.ReadLine();
                        Console.WriteLine("\n Write the Contents to the File: \n");
                        string r = Console.ReadLine();
                        using (StreamWriter fileStr = File.CreateText(fileName))
                        {
                            fileStr.WriteLine(r);
                        }
                        Console.WriteLine("File is Created...");
                        break;

                    case 2:
                        Console.Write("\n Enter the file name:");
                        fileName = Console.ReadLine();
                        if (File.Exists(fileName))
                        {
                            Console.WriteLine("File exists...");
                        }
                        else
                        {
                            Console.WriteLine("File does not exist in the current directory!");
                        }
                    }
                }
            }
        }
    }
}
```



```

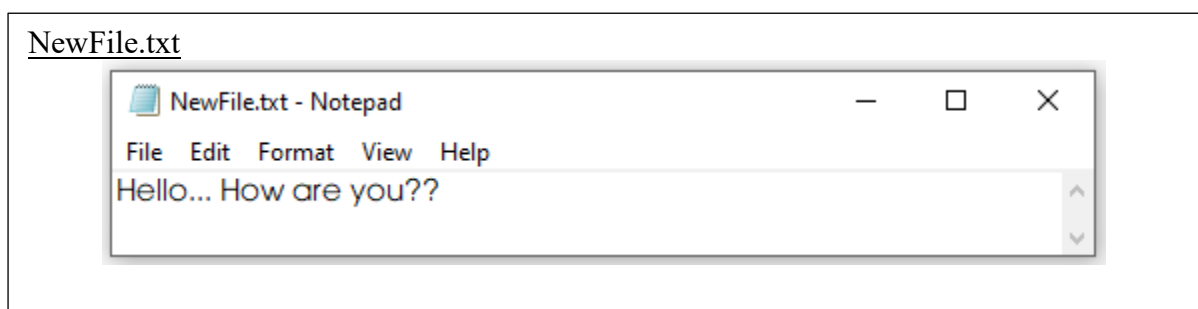
    }
    break;

case 3:
    Console.WriteLine("Enter the file name to read the contents:\n");
    fileName = Console.ReadLine();
    if (File.Exists(fileName))
    {
        using (StreamReader sr = File.OpenText(fileName))
        {
            string s = "";
            Console.WriteLine(" Here is the content of the file : ");
            while ((s = sr.ReadLine()) != null)
            {
                Console.WriteLine(s);
            }
            Console.WriteLine("");
        }
    }
    else
    {
        Console.WriteLine("File does not exists");
    }
    break;

case 4:
    Console.WriteLine("\n Exiting...");
    return;

default:
    Console.WriteLine("\n Invalid choice");
    break;
}
}
}
}
}
}

```



OUTPUT

```
E:\CSharp>csc FileRead.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>FileRead

----- MENU -----

1.Create a File
2.Existence of the File
3.Read the contents of the File
4.Exit

Enter your choice : 1

Enter the file name to create: NewFile.txt

Write the Contents to the File:

Hello... How are you??
File is Created.

----- MENU -----

1.Create a File
2.Existence of the File
3.Read the contents of the File
4.Exit

Enter your choice : 2

Enter the file name:NewFile.txt
File exists...
```

```
----- MENU -----  
  
1.Create a File  
2.Existence of the File  
3.Read the contents of the File  
4.Exit  
  
Enter your choice : 3  
Enter the file name to read the contents:  
NewFile.txt  
Here is the content of the file :  
Hello... How are you??  
  
----- MENU -----  
  
1.Create a File  
2.Existence of the File  
3.Read the contents of the File  
4.Exit  
  
Enter your choice : 4  
  
Exiting...
```

12. C# Program to Perform File Comparison.

```
using System;
using System.IO;

namespace Exercises
{
    class FileRead
    {
        public static void Main()
        {
            string file1;
            string file2;

            Console.Write("Enter the first file path:");
            file1 = Console.ReadLine();

            Console.Write("Enter the second file path:");
            file2 = Console.ReadLine();

            if (!File.Exists(file1))
            {
                Console.WriteLine("First file does not exist!");
            }

            else if (!File.Exists(file2))
            {
                Console.WriteLine("Second file does not exist!");
            }

            else if (File.ReadAllText(file1) == File.ReadAllText(file2))
            {
                Console.WriteLine("Both files contain the same content");
            }

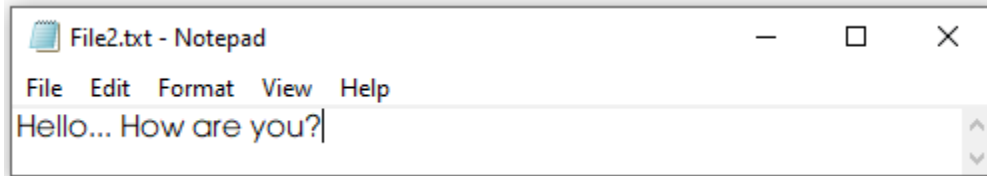
            else
            {
                Console.WriteLine("Contents of files are not same");
            }

        }
    }
}
```

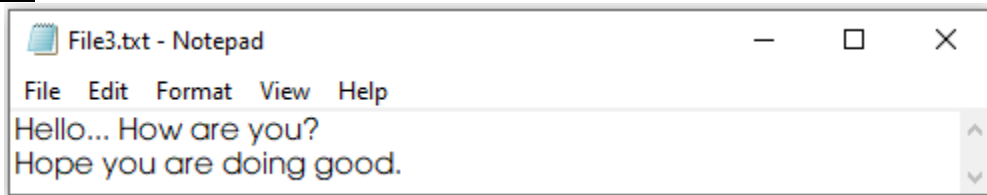
File1.txt



File2.txt



File3.txt



OUTPUT

```
E:\CSharp>csc FileCompare.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>FileCompare
Enter the first file path: E:\CSharp\File1.txt
Enter the second file path: E:\CSharp\File2.txt
Both files contain the same content

E:\CSharp>FileCompare
Enter the first file path: E:\CSharp\File1.txt
Enter the second file path: E:\CSharp\File3.txt
Contents of files are not same
```

13. C# Program to Implement IComparable Interface.

using System;

namespace Exercises

```
{
    class Fraction : IComparable
    {
        int z, n;

        public Fraction(int z, int n)
        {
            this.z = z;
            this.n = n;
        }

        public static Fraction operator +(Fraction a, Fraction b)
        {
            return new Fraction(a.z * b.n + a.n * b.z, a.n * b.n);
        }

        public static Fraction operator *(Fraction a, Fraction b)
        {
            return new Fraction(a.z * b.z, a.n * b.n);
        }

        public int CompareTo(object obj)
        {
            Fraction f = (Fraction)obj;
            if ((float)z / n < (float)f.z / f.n)
                return -1;
            else if ((float)z / n > (float)f.z / f.n)
                return 1;
            else
                return 0;
        }

        public override string ToString()
        {
            return z + "/" + n;
        }
    }

    class ICompInterface
    {
        public static void Main()
        {

```

```

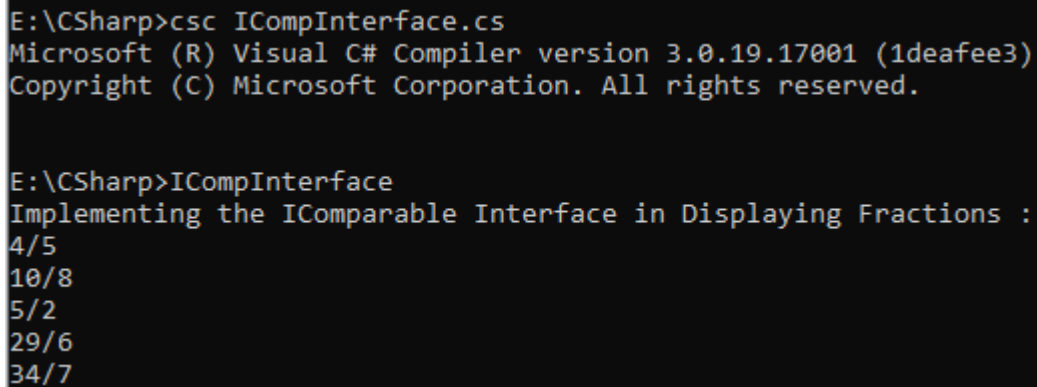
        Fraction[] a = {
            new Fraction(5,2),
            new Fraction(29,6),
            new Fraction(4,5),
            new Fraction(10,8),
            new Fraction(34,7)
        };

        Array.Sort(a);
        Console.WriteLine("Implementing the IComparable Interface in " + "Displaying
Fractions : ");
        foreach (Fraction f in a)
        {
            Console.WriteLine(f + " ");
        }

        Console.WriteLine();
        Console.ReadLine();
    }
}
}

```

OUTPUT



```

E:\CSharp>csc ICompInterface.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafee3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>ICompInterface
Implementing the IComparable Interface in Displaying Fractions :
4/5
10/8
5/2
29/6
34/7

```

14. C# Program to Create Thread Pools.

```
using System;
using System.Threading;

namespace Exercises
{
    class ThreadPoolProg
    {
        public void ThreadFun1(object obj)
        {
            int loop = 0;
            for (loop = 0; loop <= 4; loop++)
            {
                Console.WriteLine("Thread1 is executing");
            }
        }

        public void ThreadFun2(object obj)
        {
            int loop = 0;
            for (loop = 0; loop <= 4; loop++)
            {
                Console.WriteLine("Thread2 is executing");
            }
        }

        public static void Main()
        {
            ThreadPoolProg TP = new ThreadPoolProg();
            for (int i = 0; i < 2; i++)
            {
                ThreadPool.QueueUserWorkItem(new WaitCallback(TP.ThreadFun1));
                ThreadPool.QueueUserWorkItem(new WaitCallback(TP.ThreadFun2));
            }
            Console.ReadKey();
        }
    }
}
```


OUTPUT

```
E:\CSharp>csc ThreadPoolProg.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafec3)
Copyright (C) Microsoft Corporation. All rights reserved.
```

```
E:\CSharp>ThreadPoolProg
Thread2 is executing
Thread1 is executing
Thread2 is executing
Thread1 is executing
Thread2 is executing
Thread1 is executing
Thread1 is executing
Thread1 is executing
Thread1 is executing
Thread2 is executing
Thread2 is executing
Thread1 is executing
Thread2 is executing
Thread2 is executing
Thread2 is executing
Thread2 is executing
Thread2 is executing
Thread1 is executing
Thread1 is executing
Thread1 is executing
```

15. C# program to demonstrate error handling using Try, Catch and Finally block.

```
using System;
namespace Exercises
{
    class ExceptionHandling
    {
        static void Main(string[] args)
        {
            Age a = new Age();
            try
            {
                a.displayAge();
            }
            catch (AgeIsNegativeException e)
            {
                Console.WriteLine("AgeIsNegativeException: {0}", e.Message);
            }
            finally
            {
                Console.WriteLine("Execution of Finally block is done.");
            }
        }
    }
}

public class AgeIsNegativeException : Exception
{
    public AgeIsNegativeException(string message) : base(message)
    {
    }
}

public class Age
{
    int age = -5;
    public void displayAge()
    {
        if (age < 0)
        {
            throw (new AgeIsNegativeException("Age cannot be negative"));
        }
        else
        {
            Console.WriteLine("Age is: {0}", age);
        }
    }
}
```

OUTPUT

```
E:\CSharp>csc ExceptionHandling.cs
Microsoft (R) Visual C# Compiler version 3.0.19.17001 (1deafef3)
Copyright (C) Microsoft Corporation. All rights reserved.

E:\CSharp>ExceptionHandling
AgeIsNegativeException: Age cannot be negative
Execution of Finally block is done.
```

PART B – WINDOWS APPLICATION

16. C# Program to Convert Digits to Words.

Form1.cs [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 WinFormsApp1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            lbl_words.Text = NumtoWord(long.Parse(txt_num.Text));
        }

        public string NumtoWord(long number)
        {
            string word = "";
            if (number == 0)
            {
                return "Zero";
            }
            if (number < 0)
            {
                return "Minus" + Math.Abs(number);
            }
            if (number / 10000000 > 0)
            {
                word += NumtoWord(number / 10000000) + "Corer";
                number %= 10000000;
            }
        }
    }
}
```

```

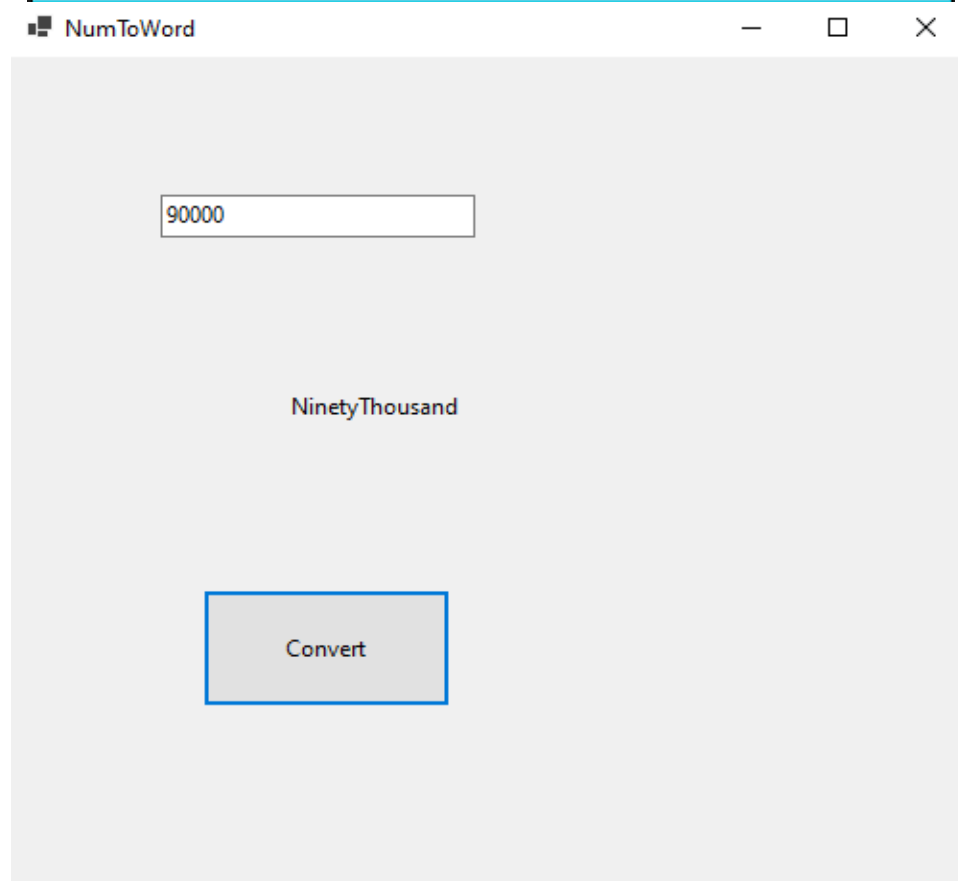
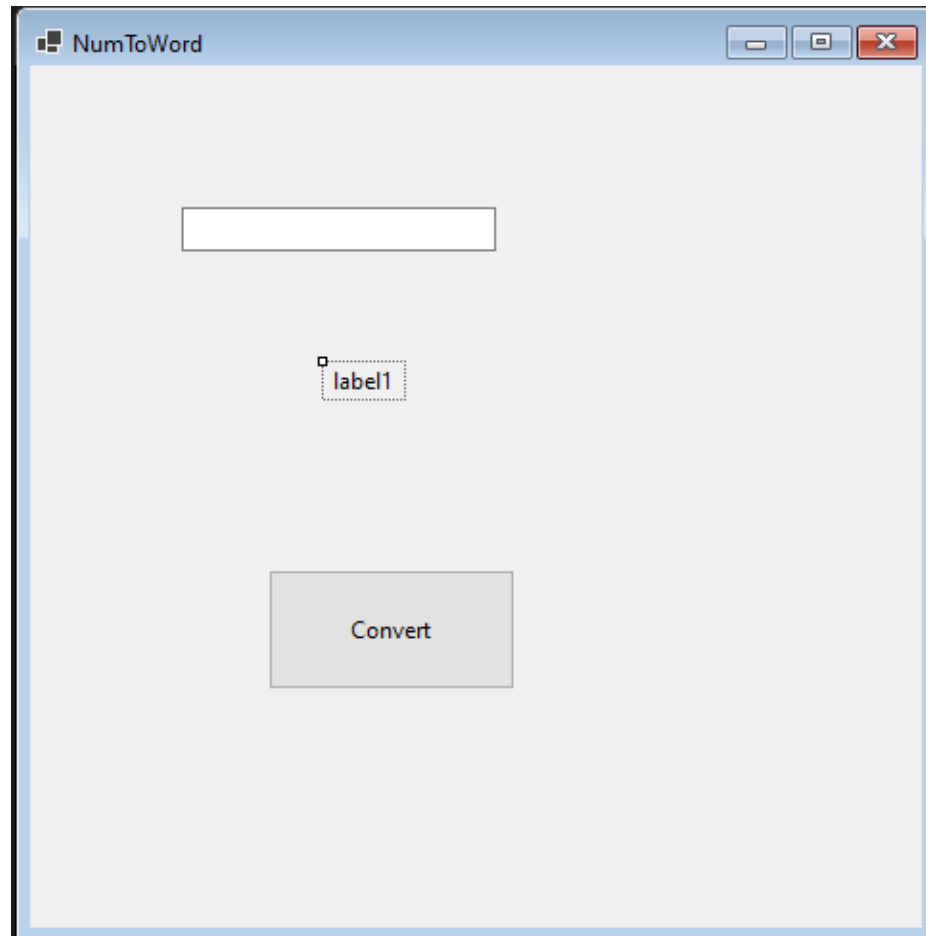
    if (number / 100000 > 0)
    {
        word += NumtoWord(number / 100000) + "Lacs";
        number %= 100000;

    }
    if (number / 1000 > 0)
    {
        word += NumtoWord(number / 1000) + "Thousand";
        number %= 1000;

    }
    if (number / 100 > 0)
    {
        word += NumtoWord(number / 100) + "Hundred";
        number %= 100;
    }
    if (number > 0)
    {
        string[] units = new string[] { "Zero", "One", "Two", "Three", "Four", "Five", "Six",
"Seven", "Eight", "Nine", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen",
"Seventeen", "Eighteen", "Nineteen" };
        string[] Tens = new string[] { "Zero", "Ten", "Twenty", "Thirty", "Fourty", "Fifty",
"Sixty", "Seventy", "Eighty", "Ninety" };
        if (number < 20)
        {
            word += units[number];
        }
        else
        {
            word += Tens[number / 10];
            if (number % 10 > 0)
            {
                word += units[number % 10];
            }
        }
    }
    return word;
}
}
}

```

OUTPUT



17. C# Program to Implement PhoneBook.

```
using System;
using System.Data;
using System.Linq;
using System.IO;
using System.Windows.Forms;

namespace PhoneBook
{

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

        private void btnNew_Click(object sender, EventArgs e)
        {
            try
            {
                panel1.Enabled = true;
                //Add a New Row:
                App.PhoneBook.AddPhoneBookRow(App.PhoneBook.NewPhoneBookRow());
                phoneBookBindingSource.MoveLast();
                txtName.Focus();
            }
            catch (Exception ex)
            {
                MessageBox.Show (ex.Message, "Message", MessageBoxButtons.OK,
                MessageBoxIcon.Error);
                App.PhoneBook.RejectChanges();
            }
        }

        private void btnEdit_Click(object sender, EventArgs e)
        {
            panel1.Enabled = true;
            txtPhone.Focus();
        }

        private void btnCancel_Click(object sender, EventArgs e)
        {
            phoneBookBindingSource.ResetBindings(false);
        }
    }
}
```

```

        panel1.Enabled = false;
    }

    private void btnSave_Click(object sender, EventArgs e)
    {
        try
        {
            //End Edit, Save Data To file:
            phoneBookBindingSource.EndEdit();
            App.PhoneBook.AcceptChanges();
            App.PhoneBook.WriteXml(string.Format("{0}//data.dat",
Application.StartupPath));
            panel1.Enabled = false;

            MessageBox.Show("Number Store Successfully:");
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Message", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            App.PhoneBook.RejectChanges();
        }
    }

    static PhoneData db;
    protected static PhoneData App
    {
        get
        {
            if (db == null)
            {
                db = new PhoneData();
            }
            return db;
        }
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        string filename = string.Format("{0}//data.dat", Application.StartupPath);
        if (File.Exists(filename))
        {
            App.PhoneBook.ReadXml(filename);
        }
        phoneBookBindingSource.DataSource = App.PhoneBook;
    }

```



```

        panel1.Enabled = false;
    }

    //Code for "DataGridView".
    private void dataGridView1_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Delete)
        {
            if (MessageBox.Show("Are You sure that you want to Delete this Record?",
"Message",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                phoneBookBindingSource.RemoveCurrent();
            }
        }
    }

    //code for "Search box";
    private void txtSearch_KeyPress_1(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar == (char)13) //enter Key:
        {
            if (!string.IsNullOrEmpty(txtSearch.Text))
            {
                //we can use linq to Query data:
                var query = from o in App.PhoneBook
                    where o.PhoneNo == txtSearch.Text
                    ||
o.FullName.ToLowerInvariant().Contains(txtSearch.Text.ToLowerInvariant())
                    || o.Email.ToLowerInvariant() == txtSearch.Text.ToLowerInvariant()
                    select o;
                dataGridView1.DataSource = query.ToList();
            }
            else
            {
                dataGridView1.DataSource = phoneBookBindingSource;
            }
        }
    }
}

```

OUTPUT

Phone Book

Phone Book

Full Name:

Ashwini

Phone No:

9988776655

Email:

ashwini@gmail.com

Address:

Hubli Kamatak

New

Edit

Cancel

Save

Search

	PhoneNo	FullName	Email	Address
▶	9988776655	Ashwini	ashwini@gmail.com	Hubli Kamatak
	8899776655	Puneet	puneet@gmail.com	Hubli Kamatak
	7788996655	Kartik	kartik@gmail.com	Manglore kamata...
	6677889955	Praveen	praveen2gmail.com	Savadatti Kamat...
	5566778899	Sanjana	sanjana@gmail.com	Savadatti Kamat...

Phone Book

Phone Book

Full Name:

Sanjana

Phone No:

5566778899

Email:

sanjana@gmail.com

Address:

Savadatti Kamataka

New

Edit

Cancel

Save

Search

Puneet

	PhoneNo	FullName	Email	Address
▶	8899776655	Puneet	puneet@gmail.com	Hubli Kamatak

18. C# Program to Perform Reversal, Padding and Trimming Operations on string.

```
using System;
using System.Windows.Forms;

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

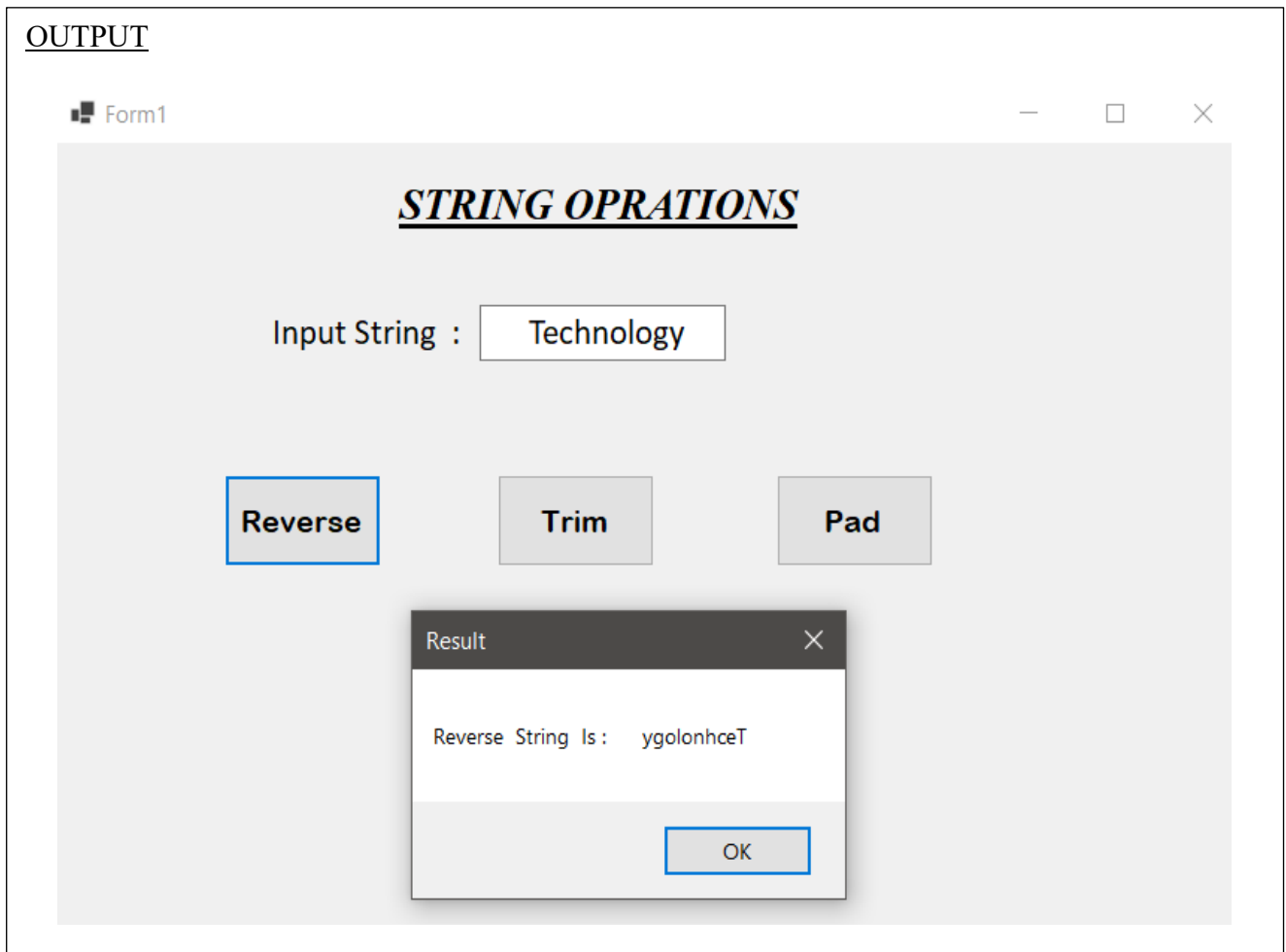
        private void btnrev_Click(object sender, EventArgs e)
        {
            string inputString, revstr = "";
            int Length;
            inputString = txtInput.Text;
            Length = inputString.Length-1;
            while (Length >= 0)
            {
                revstr = revstr + inputString[Length];
                Length--;
            }

            MessageBox.Show("Reverse String Is : " + revstr, "Result");
        }

        private void btntrim_Click(object sender, EventArgs e)
        {
            string inputString;
            inputString = txtInput.Text;
            MessageBox.Show("The String After Trimming : " + inputString.Trim(), "Result");
        }

        private void btnpad_Click(object sender, EventArgs e)
        {
            string inputString;
            inputString = txtInput.Text;
            inputString = inputString.PadLeft(10, '*');
            inputString = inputString.PadRight(15, '*');
            MessageBox.Show("String After Padding : " + inputString, "Result");
        }
    }
}
```

OUTPUT



19. C# Program to Create a Progress Bar Control.

```
using System;
using System.ComponentModel;
using System.Threading;
using System.Windows.Forms;

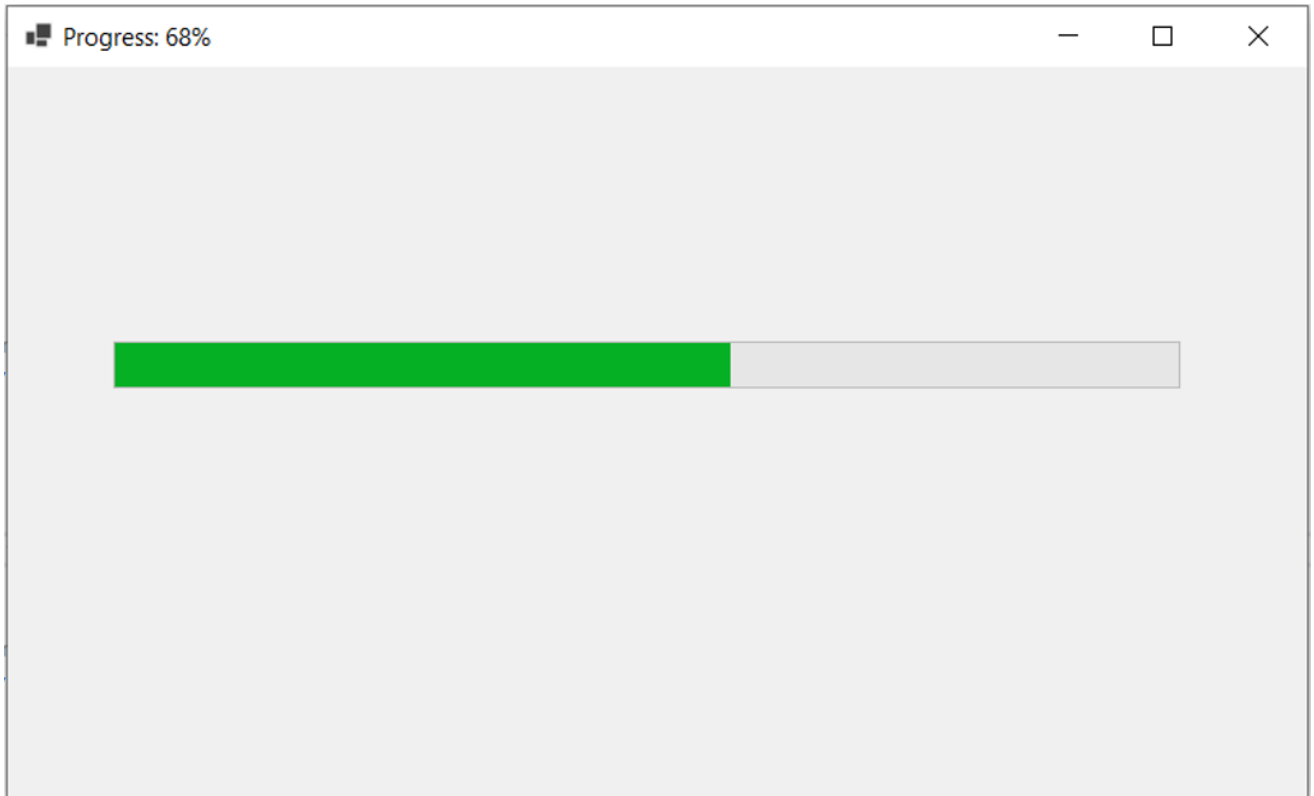
namespace WindowsFormsApplication1
{
    public partial class Form1: Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }

    private void Form1_Load(object sender, System.EventArgs e)
    {
        backgroundWorker1.WorkerReportsProgress = true;
        backgroundWorker1.RunWorkerAsync();
    }

    private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
    {
        for (int i = 1; i <= 100; i++)
        {
            Thread.Sleep(50);
            backgroundWorker1.ReportProgress(i);
        }
    }

    private void backgroundWorker1_ProgressChanged(object sender,
        ProgressChangedEventArgs e)
    {
        progressBar1.Value = e.ProgressPercentage;
        this.Text = "Progress: " + e.ProgressPercentage.ToString() + "%";
    }
}
```

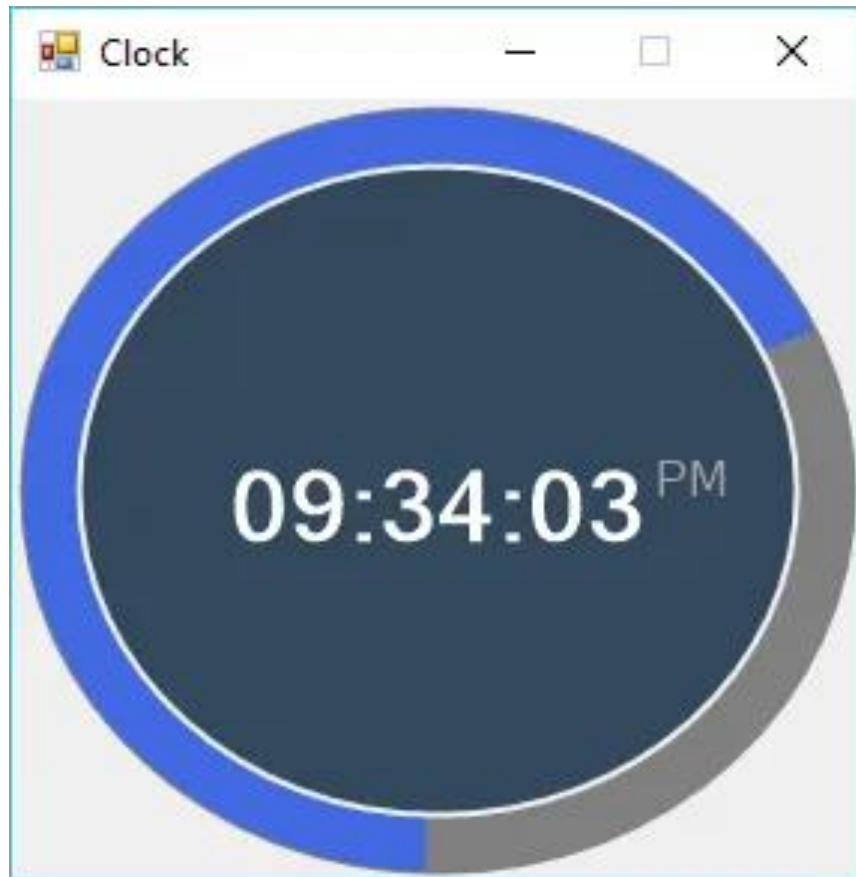
OUTPUT



20. Develop a winform application to create flat clock.

```
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 clock
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            timer1.Start();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            System.Timers.Timer timer = new System.Timers.Timer();
            timer.Interval = 1000;//1s
            timer.Elapsed += Timer_Elapsed;
            timer.Start();
        }
        private void Timer_Elapsed(object sender, System.Timers.ElapsedEventArgs e)
        {
            circularProgressBar1.Invoke((MethodInvoker)delegate
            {
                circularProgressBar1.Text = DateTime.Now.ToString("hh:mm:ss");
                circularProgressBar1.SubscriptText = DateTime.Now.ToString("tt");//AM or PM
            });
        }
    }
}
```

OUTPUT



21. C# Program to perform a number guessing game.

```
using System;
using System.Drawing;
using System.Windows.Forms;
namespace game
{
    public class Form1 : Form
    {
        // Initialising component
        static Random r = new Random();
        int value;
        int guessnum;
        int win = 10;
        int guess = 1;
        Button button1;
        TextBox textBox1;
        RichTextBox richTextBox1;
        RichTextBox richTextBox2;
        Label label4;

        public Form1()
        {
            InitializeComponent();
        }

        void InitializeComponent()
        {
            value = r.Next(100);
            this.Controls.Clear();
            this.BackColor = Color.SkyBlue;
            this.AutoSize = true;
            this.Padding = new Padding( 16 );

            Label label = new Label();
            label.Text = "Pick a number between 1 and 100";
            label.Bounds = new Rectangle( 10, 20, 340, 40 );
            label.Font = new Font( "Arial", 16 );

            textBox1 = new TextBox();
            textBox1.Bounds = new Rectangle( 20, 50, 120, 80 );
            textBox1.Font = new Font( "Arial", 24 );

            button1 = new Button();
            button1.Text = " Check Your Guess ";
            button1.Bounds = new Rectangle( 160, 50, 120, 40 );
```

```

button1.BackColor = Color.LightGray;
button1.Click += new EventHandler( button1_Click );

Label label2 = new Label();
label2.Text = "Low Guess";
label2.Bounds = new Rectangle( 20, 150, 160, 40 );
label2.Font = new Font( "Arial", 18 );

richTextBox1 = new RichTextBox();
richTextBox1.Bounds = new Rectangle( 20, 190, 160, 300 );
richTextBox1.Font = new Font( "Arial", 16 );

Label label3 = new Label();
label3.Text = "High Guess";
label3.Bounds = new Rectangle( 180, 150, 160, 40 );
label3.Font = new Font( "Arial", 18 );

richTextBox2 = new RichTextBox();
richTextBox2.Bounds = new Rectangle( 180, 190, 160, 300 );
richTextBox2.Font = new Font( "Arial", 16 );

label4 = new Label();
label4.Bounds = new Rectangle( 20, 100, 340, 40 );
label4.Font = new Font( "Arial", 16 );

this.Controls.Add( label );
this.Controls.Add( textBox1 );
this.Controls.Add( button1 );
this.Controls.Add( label4 );
this.Controls.Add( label2 );
this.Controls.Add( label3 );
this.Controls.Add( richTextBox1 );
this.Controls.Add( richTextBox2 );
}

private void button1_Click(object sender,EventArgs e)
{
    // Coding of game
    if( textBox1.Text == "" )
    {
        return;
    }
    guessnum = Convert.ToInt32(textBox1.Text);
    textBox1.Text = String.Empty;
    if (win >= 0)
    {

```

```

        if (guessnum == value)
        {
            MessageBox.Show( "You have guessed the number! \n The number was " + value );
            InitializeComponent();
        }
        else if (guessnum < value)
        {
            richTextBox1.Text += guessnum + "\n";
            label4.Text = "wrong Guess and number of guesses left are " + (10 - guess);
        }
        else if (guessnum > value)
        {
            richTextBox2.Text += guessnum + "\n";
            label4.Text = "wrong Guess and number of guesses left are " + (10 - guess);
        }
        guess++;
        win--;
    }
    if (guess == 11)
    {
        label4.Text = "You loose,Correct Guess is " + value;
    }
}

static void Main()
{
    Application.Run( new Form1() );
}
}

```

OUTPUT

Pick a number between 1 and 100

wrong Guess and number of guesses left are 7

Low Guess	High Guess
55	
65	
75	

Pick a number between 1 and 100

wrong Guess and number of guesses left are 6

Low Guess	High Guess
55	85
65	
75	

Pick a number between 1 and 100

wrong Guess and number of guesses left are 1

Low Guess	High Guess
55	85
65	80
75	81
70	82
	83

Pick a number between 1 and 100

wrong Guess and number of guesses left are 0

Low Guess	High Guess
55	
65	
75	
70	
	83
	84

You loose, Correct Guess is 76

22. Develop an application to create a notepad.

```
using System;
using System.IO;
using System.Windows.Forms;
using System.Drawing;

namespace Notepad
{
    class NotepadForm : Form
    {
        private string fileName;
        private RichTextBox txtContent;
        private ToolBar toolBar;

        internal NotepadForm()
        {
            fileName = null;
            initializeComponents();
        }

        void initializeComponents()
        {
            this.Text = "My notepad";
            this.MinimumSize = new Size( 600, 450 );
            this.FormClosing += new FormClosingEventHandler( NotepadClosing );
            this.MaximizeBox = true;

            toolBar = new ToolBar();
            toolBar.Font = new Font( "Arial", 16 );
            toolBar.Padding = new Padding( 4 );
            toolBar.ButtonClick += new ToolBarButtonClickEventHandler(toolBarClicked);

            ToolBarButton toolBarButton1 = new ToolBarButton();
            ToolBarButton toolBarButton2 = new ToolBarButton();
            ToolBarButton toolBarButton3 = new ToolBarButton();
            toolBarButton1.Text = "New";
            toolBarButton2.Text = "Open";
            toolBarButton3.Text = "Save";

            toolBar.Buttons.Add(toolBarButton1);
            toolBar.Buttons.Add(toolBarButton2);
            toolBar.Buttons.Add(toolBarButton3);

            txtContent = new RichTextBox();
            txtContent.Size = this.ClientSize;
            txtContent.Height -= toolBar.Height;
            txtContent.Top = toolBar.Height;
            txtContent.Anchor = AnchorStyles.Left | AnchorStyles.Right |
AnchorStyles.Top | AnchorStyles.Bottom;
            txtContent.Font = new Font( "Arial", 16 );
            txtContent.AcceptsTab = true;
            txtContent.Padding = new Padding( 8 );
        }
    }
}
```

```

        this.Controls.Add(toolBar);
        this.Controls.Add( txtContent );
    }

    private void toolBarClicked (Object sender, ToolBarButtonClickEventArgs e)
    {
        saveFile();

        switch (toolBar.Buttons.IndexOf(e.Button))
        {
            case 0: this.Text += "My notepad";
                    txtContent.Text = string.Empty;
                    fileName = null;
                    break;

            case 1: OpenFileDialog openDlg = new OpenFileDialog();
                    if (DialogResult.OK == openDlg.ShowDialog())
                    {
                        fileName = openDlg.FileName;
                        txtContent.LoadFile( fileName );
                        this.Text = "My notepad " + fileName;
                    }
                    break;
        }
    }

    void saveFile()
    {
        if( fileName == null )
        {
            SaveFileDialog saveDlg = new SaveFileDialog();

            if (DialogResult.OK == saveDlg.ShowDialog())
            {
                fileName = saveDlg.FileName;
                this.Text += " " + fileName;
            }
        }
        else
        {
            txtContent.SaveFile( fileName, RichTextBoxStreamType.RichText );
        }
    }

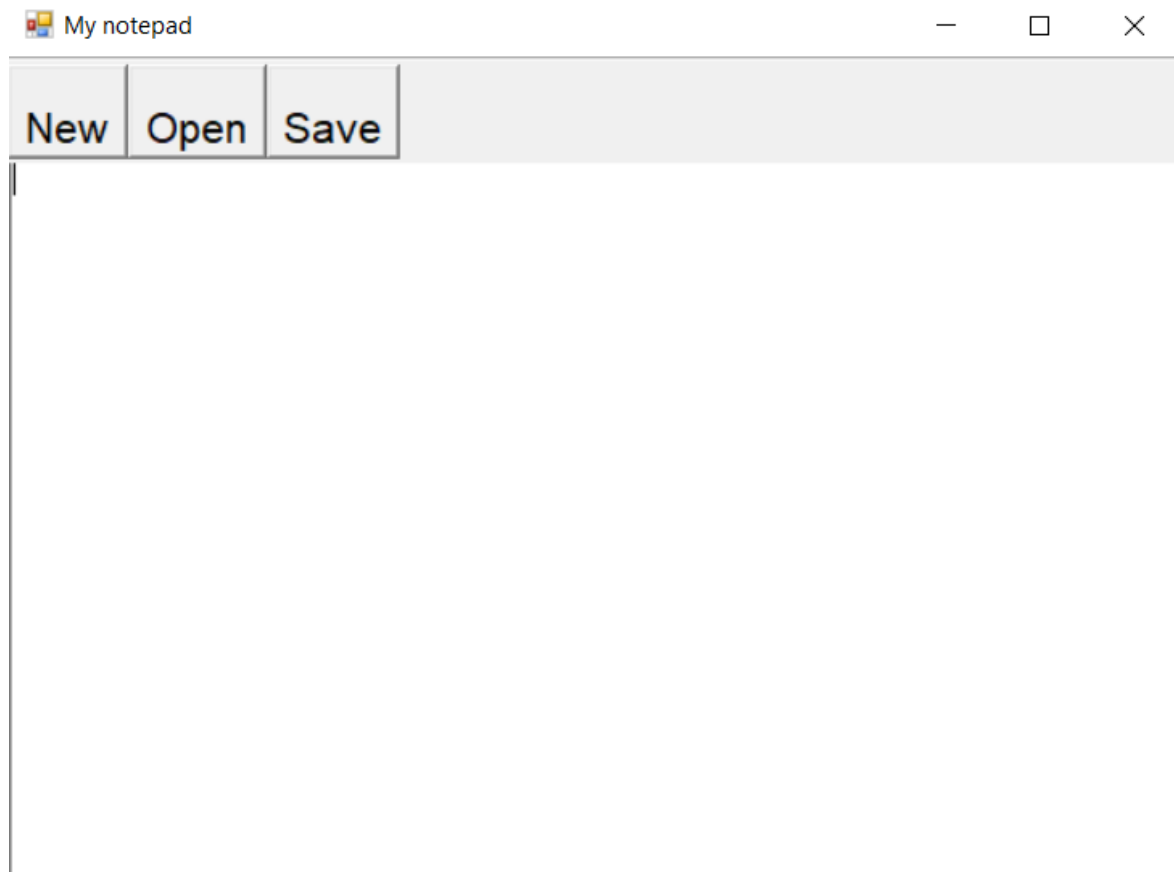
    private void NotepadClosing(Object sender, FormClosingEventArgs e)
    {
        saveFile();
    }

    static void Main (String[] args )
    {

```

```
        Application.Run( new NotepadForm() );  
    }  
}
```

OUTPUT



23. C# Program to Perform Reversal, Padding and Trimming Operations on string.

```
using System;
using System.Windows.Forms;
using System.Collections.Generic;
using System.Drawing;
using System.Drawing.Drawing2D;

namespace BinaryTree
{
    partial class BinTreeForm : Form
    {
        private Node root;

        BinTreeForm()
        {
            InitializeComponent();
            this.root = null;
            test();
        }

        void test()
        {
            textBox1.Text = "5";
            btnAdd_Click( btnAdd, null );
            textBox1.Text = "3";
            btnAdd_Click( btnAdd, null );
            textBox1.Text = "2";
            btnAdd_Click( btnAdd, null );
            textBox1.Text = "1";
            btnAdd_Click( btnAdd, null );
            textBox1.Text = "4";
            btnAdd_Click( btnAdd, null );
            textBox1.Text = "7";
            btnAdd_Click( btnAdd, null );
            textBox1.Text = "6";
            btnAdd_Click( btnAdd, null );
            textBox1.Text = "8";
            btnAdd_Click( btnAdd, null );
        }

        void btnCreate_Click( object sender, EventArgs e )
        {
            root = null;
            pictureBox1.Image = null;
        }
    }
}
```



```

void btnAdd_Click( object sender, EventArgs e )
{
    int value = int.Parse( textBox1.Text );
    if( root == null )
        root = new Node( value );
    else
    {
        if( root.Add( value ) == false )
            MessageBox.Show( "The value already exists!" );
    }
    drawTree();
}

void btnRemove_Click( object sender, EventArgs e )
{
    int value = int.Parse( textBox1.Text );
    if( root != null )
    {
        bool status = root.Remove( value, root, ref root );
        if( status == false )
        {
            MessageBox.Show( "the value does not exists" );
        }
    }
    drawTree();
}

void btnSearch_Click( object sender, EventArgs e )
{
    string msg;
    int value = int.Parse( textBox1.Text );
    if ( root == null )
    {
        msg = "Tree is empty";
    } else
    {
        if( root.Exists( value ) )
        {
            msg = "Value found";
        } else
        {
            msg = "Value not found";
        }
    }
    MessageBox.Show(msg);
}

```

```

void drawTree()
{
    if( root != null )
        pictureBox1.Image = root.Draw();
    else
        pictureBox1.Image = null;
    this.Update();
}

static void Main()
{
    Application.Run( new BinTreeForm() );
}
}

class Node
{
    internal Node left { get; set; }
    internal Node right { get; set; }
    internal int value;
    internal int center = 12;
    private static Bitmap nodeBg = new Bitmap( 30, 25 );
    private static Font font = new Font("Arial", 14);

    internal Node( int value )
    {
        this.value = value;
    }

    internal bool Add( int value )
    {
        Node node = new Node( value );
        if( value < this.value )
        {
            if( this.left == null )
            {
                this.left = node;
                return true;
            }
            else
                return this.left.Add( value );
        }
        else if ( value > this.value )
        {
            if( this.right == null )

```

```

        {
            this.right = node;
            return true;
        }
        else
            return this.right.Add( value );
    }
    return false;
}

internal bool Remove(int value, Node parent, ref Node root)
{
    if ( value < this.value )
    {
        if( left != null )
        {
            return left.Remove( value, this, ref root );
        }
    }
    else if (value > this.value)
    {
        if( right != null )
        {
            return right.Remove( value, this, ref root );
        }
    }
    else if( value == this.value )
    {
        bool isLeft = (this == parent.left );
        if( left == null && right == null )
        {
            if( root == this )
                root = null;
            else
                if (isLeft) parent.left = null; else parent.right = null;
        } else if (right == null)
        {
            if (isLeft) parent.left = left; else parent.right = left;
            if( root == this )
                root = left;
        } else
        {
            if ( right.left == null )
            {
                right.left = left;
                if (isLeft) parent.left = right;
            }
            else
            {
                if (right.right == null)
                {
                    right.right = left;
                    if (!isLeft) parent.right = right;
                }
                else
                {
                    Node temp = right.right;
                    right.right = left;
                    temp.left = right;
                    if (!isLeft) parent.right = temp;
                }
            }
        }
    }
    else

```

```

        parent.right = right;
        if( root == this )
            root = right;
    }
    else
    {
        Node node = right;
        while ( node.left.left != null )
            node = node.left;
        Console.WriteLine( "Node: " + node.value );
        this.value = node.left.value;
        Console.WriteLine( "here" );
        node.left = null;
    }
}
return true;
}

return false;
}

public Image Draw()
{
    Size lSize = new Size( nodeBg.Width / 2, 0 );
    Size rSize = new Size( nodeBg.Width / 2, 0 );
    Image lNodeImg = null;
    Image rNodeImg = null;
    int lCenter = 0, rCenter = 0;

    if ( this.left != null )
    {
        lNodeImg = left.Draw();
        lSize = lNodeImg.Size;
        this.center = lSize.Width;
        lCenter = left.center;
    }
    if ( this.right != null )
    {
        rNodeImg = right.Draw();
        rSize = rNodeImg.Size;
        rCenter = right.center;
    }

    int maxHeight = ( lSize.Height < rSize.Height ) ? rSize.Height : lSize.Height;

    if ( maxHeight > 0 ) maxHeight += 35;
}

```

```

        Size resultSize = new Size ( lSize.Width + rSize.Width, nodeBg.Size.Height +
maxHeight );
        Bitmap result = new Bitmap(resultSize.Width, resultSize.Height);

        Graphics g = Graphics.FromImage(result);
        g.SmoothingMode = SmoothingMode.HighQuality;

        g.FillRectangle(Brushes.White, new Rectangle(new Point(0, 0), resultSize));
        g.DrawImage(nodeBg, lSize.Width - nodeBg.Width / 2, 0);

        string str = "" + value;
        g.DrawString( str, font, Brushes.Black, lSize.Width - nodeBg.Width / 2 + 7,
nodeBg.Height / 2f - 12);

        Pen pen = new Pen(Brushes.Black, 1.2f);
        float x1 = center;
        float y1 = nodeBg.Height;
        float y2 = nodeBg.Height + 35;
        float x2 = lCenter;
        var h = Math.Abs(y2 - y1);
        var w = Math.Abs(x2 - x1);
        if (lNodeImg != null)
        {
            g.DrawImage(lNodeImg, 0, nodeBg.Size.Height + 35);
            var points1 = new List<PointF>
                {
                    new PointF(x1, y1),
new PointF(x1 - w/6, y1 + h/3.5f),
new PointF(x2 + w/6, y2 - h/3.5f),
new PointF(x2, y2),
                };
            g.DrawCurve(pen, points1.ToArray(), 0.5f);
        }
        if (rNodeImg != null)
        {
            g.DrawImage(rNodeImg, lSize.Width, nodeBg.Size.Height + 35);
            x2 = rCenter + lSize.Width;
            w = Math.Abs(x2 - x1);
            var points = new List<PointF>
                {
                    new PointF(x1, y1),
new PointF(x1 + w/6, y1 + h/3.5f),
new PointF(x2 - w/6, y2 - h/3.5f),
new PointF(x2, y2)
                };
        }
    }
}

```

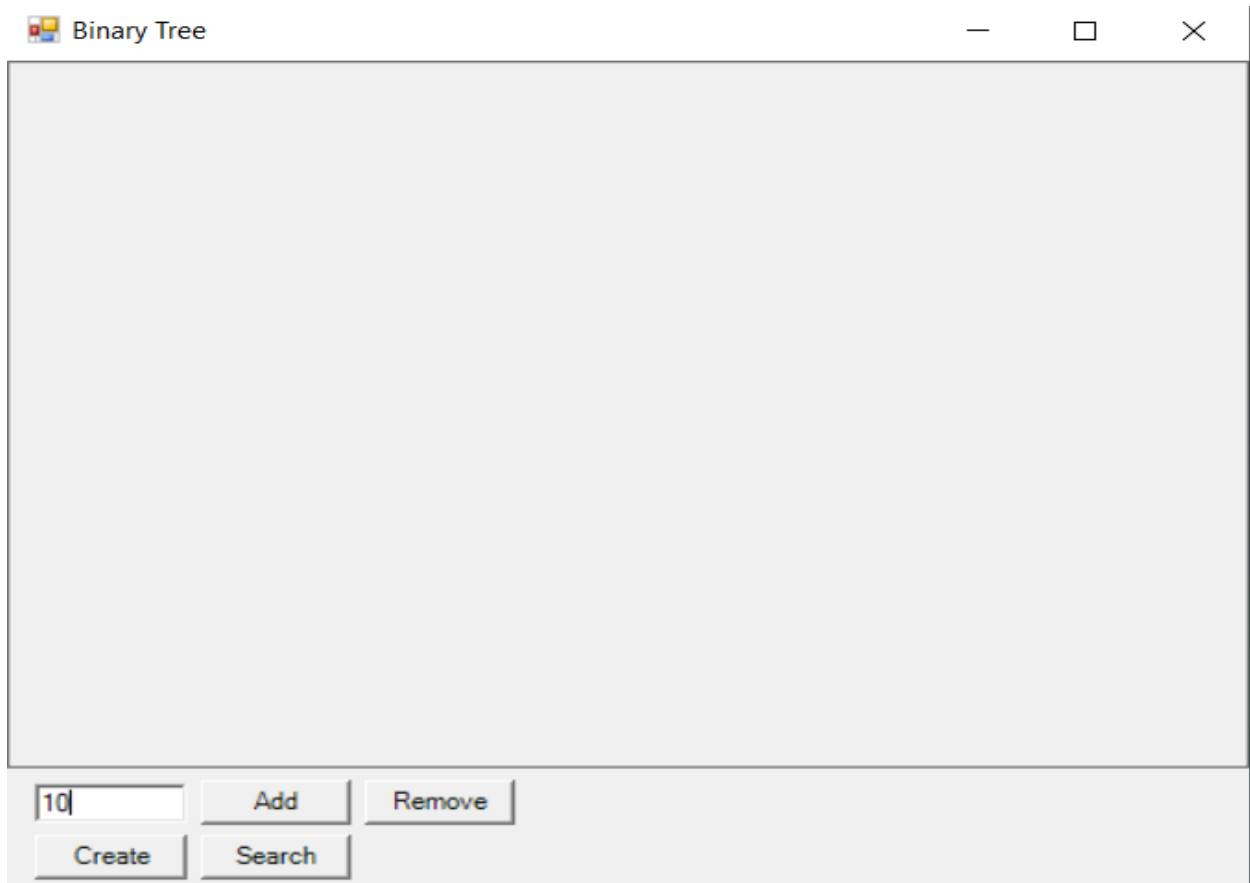
```

        g.DrawCurve(pen, points.ToArray(), 0.5f);
    }
    return result;
}

public bool Exists(int value)
{
    bool res = value == this.value;
    if (!res && left != null)
        res = left.Exists(value);
    if (!res && right != null)
        res = right.Exists(value);
    return res;
}
}
}

```

OUTPUT



OUTPUT

