

DEV SANSKRITI VISHWAVIDYALAYA



SESSION 2018-2021

PRACTICAL ON “ C# ”

Submitted to:

Mr. Chandrashekhar Patel

Submitted by:

Anuradha Kumari

BCA(5th sem)

Department of Computer Science

Dev Sanskriti Vishwavidalaya Gayatrikunj-shantikunj,Haridwar

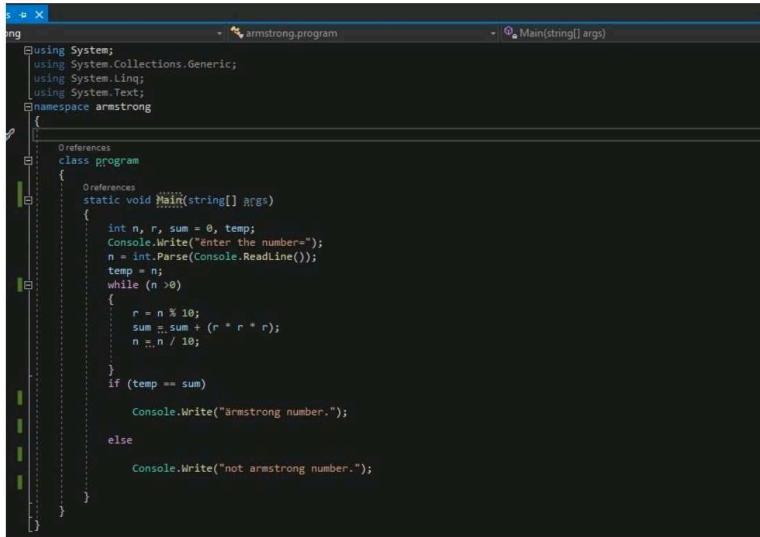
www.dsuv.ac.in

| S.no | Topics | Page no |
|------|--|---------|
| 1) | write a programme to print Armstrong number:- | 4 |
| 2 | write a programme for print factorial of a number: | 5 |
| 3) | write a programme for find the GCD of 2 numbers: | 6 |
| 4) | write a programme to check if a number is a prime number | 7 |
| 5) | write a programme to print the fibonacci series : | 8 |
| 6) | write a programme to print the half pyramid pattern: | 9 |
| 7) | Write a programme to print the half pyramid pattern with numbers: | 10 |
| 8) | Write a programme to print half pyramid inverse pattern: | 11 |
| 9) | write a programme to print the pyramid pattern: | 12 |
| 10) | Write a programme to print the inverse pyramid pattern: | 13 |
| 11) | Write a programme to print the diamond pattern: | 14 |
| 12) | Write a programme to print the pascal's triangle: | 15 |
| 13) | Write a programme to compare 2 string without using string library function: | 16 |
| 14) | Write a programme to count a total numbers of alphabets , digits and special characters in a string: | 17 |
| 15) | Write a programme to copy one string to another string: | 18 |
| 16) | Write a programme to find maximum occurring character in a string: | 19 |
| 17) | Write a programme to check whether a given substring is present in the given string: | 20 |
| 18) | Write a programme for abstraction: | 21 |
| 19) | Write a programme for single inheritance: | 22 |
| 20) | Write a programme for multi level inheritance | 23 |

| | | |
|-----|---|-------|
| 21) | Write a programme for multiple inheritance: | 24-25 |
| 22) | Write a programme for method overloading: | 26 |
| 23) | Write a programme for method overriding: | 27 |
| 24) | Write a programme for interface: | 28 |
| 25) | Write a programme for exception handling through try and catch: | 29 |
| 26) | Write a programme for properties. | 30 |
| 27) | Write a programme for threading. | 31 |
| 28) | Write a programme to access data from database using ADO.net. | 32 |
| 29) | Write a programme using namespace. | 33 |

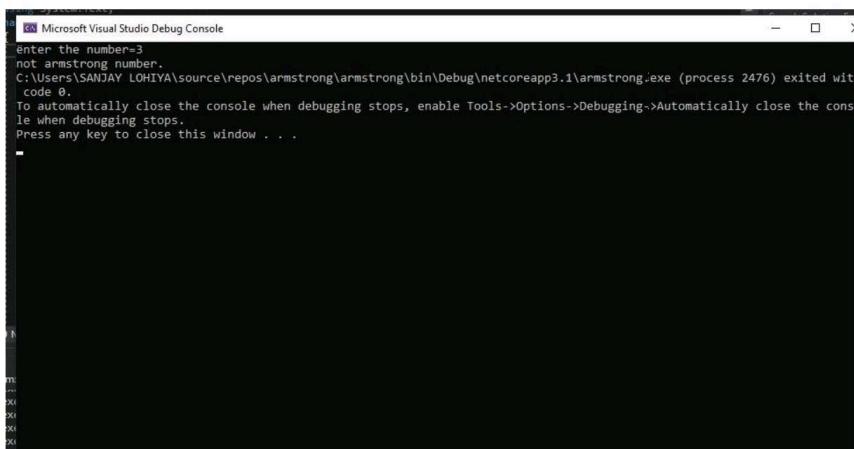
1] write a programme to print Armstrong number:-

Program:-



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace armstrong
{
    class program
    {
        static void Main(string[] args)
        {
            int n, r, sum = 0, temp;
            Console.Write("Enter the number=");
            n = int.Parse(Console.ReadLine());
            temp = n;
            while (n > 0)
            {
                r = n % 10;
                sum = sum + (r * r * r);
                n = n / 10;
            }
            if (temp == sum)
                Console.WriteLine("Armstrong number.");
            else
                Console.WriteLine("not Armstrong number.");
        }
    }
}
```

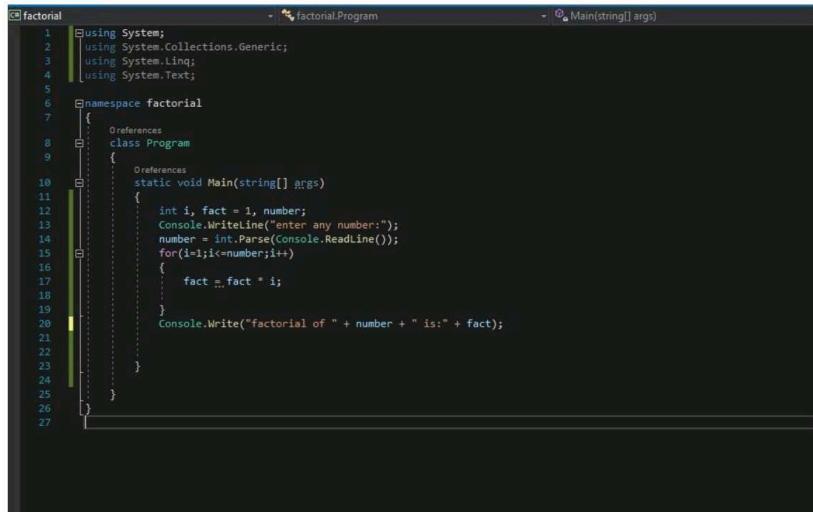
Output:-



```
Microsoft Visual Studio Debug Console
enter the number=3
not Armstrong number.
C:\Users\SANDAY LOHAYA\source\repos\armstrong\armstrong\bin\Debug\netcoreapp3.1\armstrong.exe (process 2476) exited with
code 0.
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 . . .

```

2] write a programme for print factorial of a number: Program:-



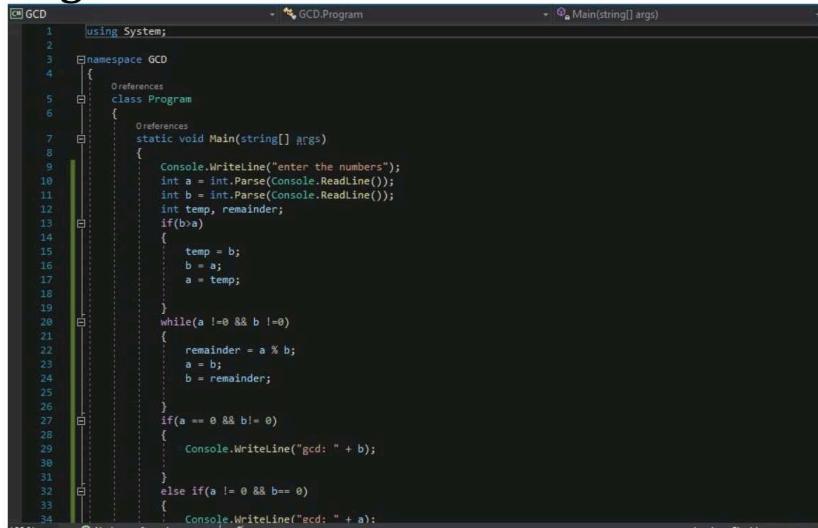
```
factorial          ↗ factorial.Program
1  Using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5
6  namespace factorial
7  {
8      class Program
9      {
10         static void Main(string[] args)
11         {
12             int i, fact = 1, number;
13             Console.WriteLine("enter any number:");
14             number = int.Parse(Console.ReadLine());
15             for(i=1;i<=number;i++)
16             {
17                 fact *= i;
18             }
19             Console.Write("factorial of " + number + " is:" + fact);
20         }
21     }
22 }
23
24
25
26
27 }
```

Output:

```
enter any number:
5
factorial of 5 is:120
C:\Users\SANDAY LOHIA\source\repos\fatorial\fatorial\bin\Debug\netcoreapp3.1\fatorial.exe (process 3416) exited with
code 0.
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 . . .
```

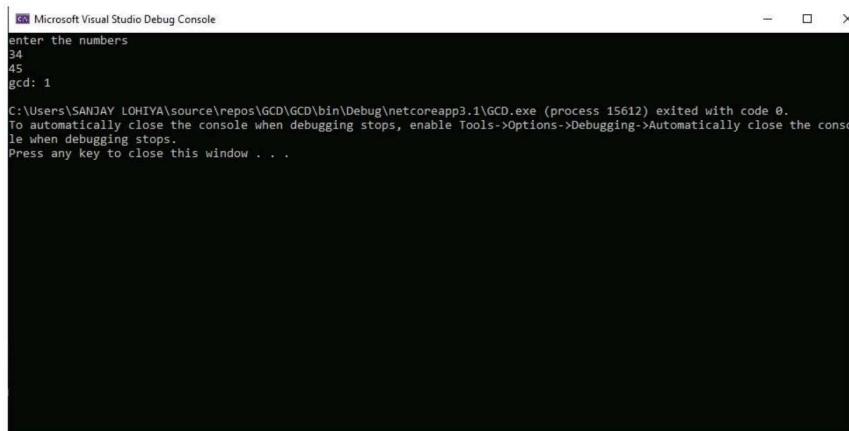
3] write a programme for find the GCD of 2 numbers:

Program



```
1  using System;
2
3  namespace GCD
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("enter the numbers");
10             int a = int.Parse(Console.ReadLine());
11             int b = int.Parse(Console.ReadLine());
12             int temp, remainder;
13             if(b>a)
14             {
15                 temp = b;
16                 b = a;
17                 a = temp;
18             }
19             while(a !=0 && b !=0)
20             {
21                 remainder = a % b;
22                 a = b;
23                 b = remainder;
24             }
25             if(a == 0 && b!= 0)
26             {
27                 Console.WriteLine("gcd: " + b);
28             }
29             else if(a != 0 && b== 0)
30             {
31                 Console.WriteLine("gcd: " + a);
32             }
33         }
34     }
```

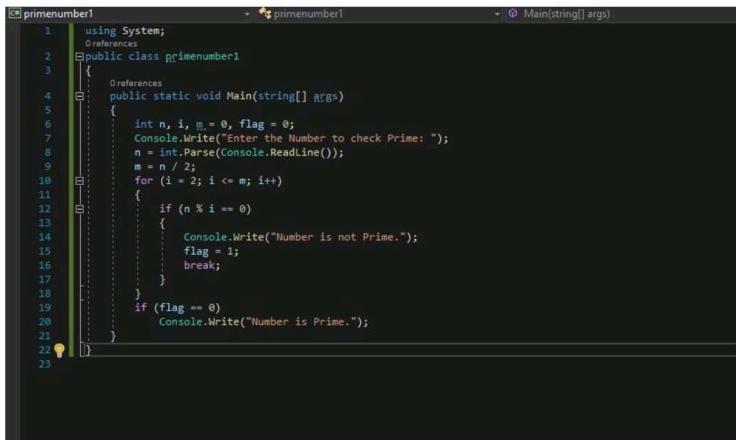
Output:



```
Microsoft Visual Studio Debug Console
enter the numbers
34
45
gcd: 1
C:\Users\SANJAY LOHIYA\source\repos\GCD\GCD\bin\Debug\netcoreapp3.1\GCD.exe (process 15612) exited with code 0.
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 . . .
```

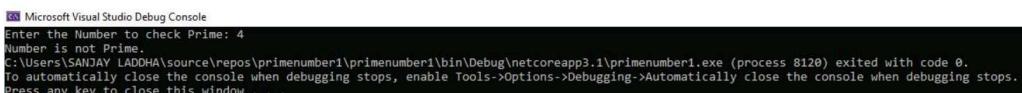
4] write a programme to check if a number is a prime number

Program:-



```
primenumber1.cs
1  using System;
2  References
3  public class primenumber1
4  {
5      References
6      public static void Main(string[] args)
7      {
8          int n, i, m = 0, flag = 0;
9          Console.WriteLine("Enter the Number to check Prime: ");
10         n = int.Parse(Console.ReadLine());
11         m = n / 2;
12         for (i = 2; i <= m; i++)
13         {
14             if (n % i == 0)
15             {
16                 Console.WriteLine("Number is not Prime.");
17                 flag = 1;
18                 break;
19             }
20             if (flag == 0)
21                 Console.WriteLine("Number is Prime.");
22         }
23     }
```

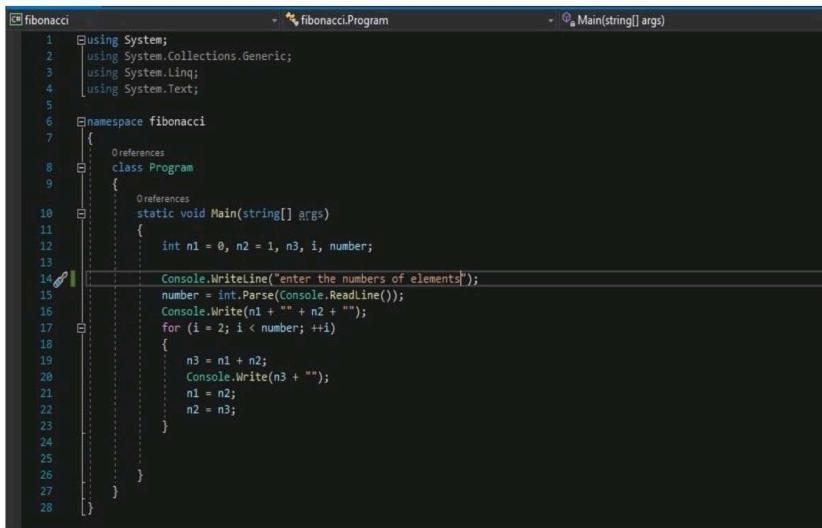
Output:



```
Microsoft Visual Studio Debug Console
Enter the Number to Check Prime: 4
Number is not Prime.
C:\Users\SANJAY LADDHA\source\repos\primenumber1\primenumber1\bin\Debug\netcoreapp3.1\primenumber1.exe (process 8120) exited with code 0.
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 . . .
```

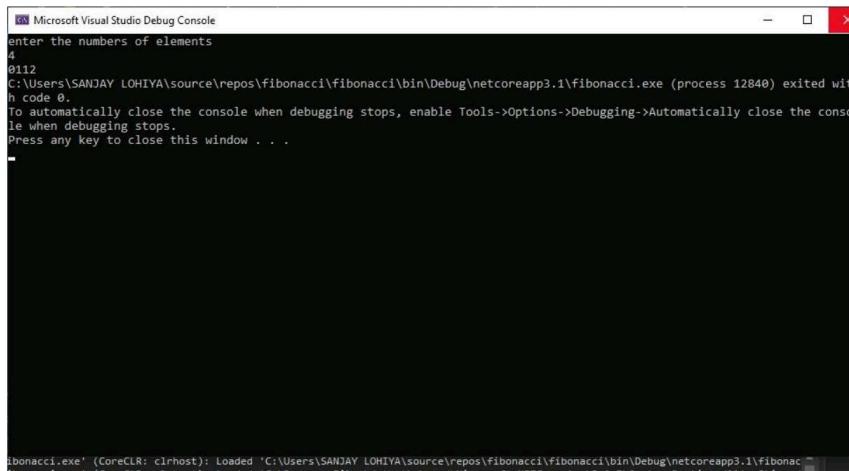
5] write a programme to print the fibonacci series :

Program:-



```
fibonacci          fibonacci.Program        Main(string[] args)
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5
6  namespace fibonacci
7  {
8      class Program
9      {
10         static void Main(string[] args)
11         {
12             int n1 = 0, n2 = 1, n3, i, number;
13
14             Console.WriteLine("enter the numbers of elements");
15             number = int.Parse(Console.ReadLine());
16             Console.Write(n1 + "" + n2 + "");
17             for (i = 2; i < number; ++i)
18             {
19                 n3 = n1 + n2;
20                 Console.Write(n3 + " ");
21                 n1 = n2;
22                 n2 = n3;
23             }
24
25
26
27         }
28     }
}
```

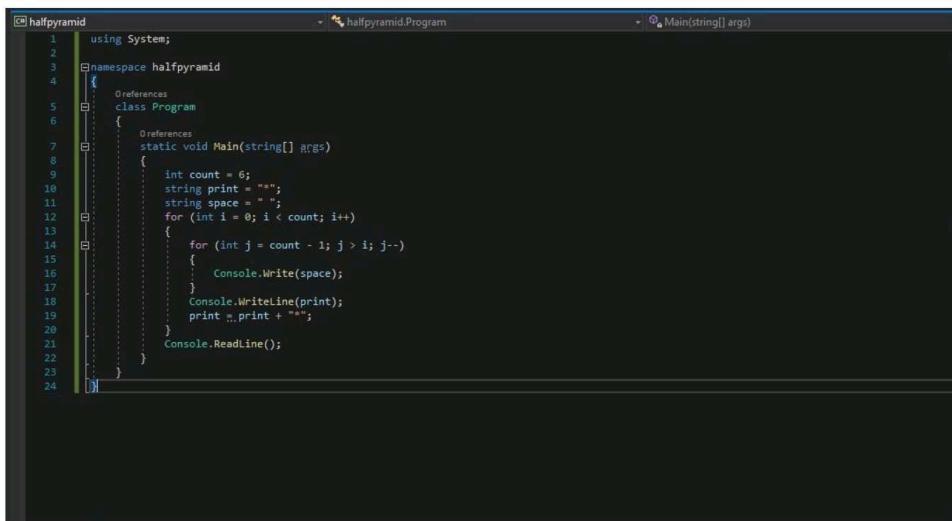
Output:



```
Microsoft Visual Studio Debug Console
enter the numbers of elements
4
0112
C:\Users\SANJAY LOHIYA\source\repos\fibonacci\fibonacci\bin\Debug\netcoreapp3.1\fibonacci.exe (process 12840) exited with
h code 0.
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 . . .
```

6] write a programme to print the half pyramid pattern:

Program:-



```
halfpyramid
1  using System;
2
3  namespace halfpyramid
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int count = 6;
10             string print = "*";
11             string space = " ";
12             for (int i = 0; i < count; i++)
13             {
14                 for (int j = count - 1; j > i; j--)
15                 {
16                     Console.Write(space);
17                 }
18                 Console.WriteLine(print);
19                 print += "*";
20             }
21             Console.ReadLine();
22         }
23     }
24 }
```

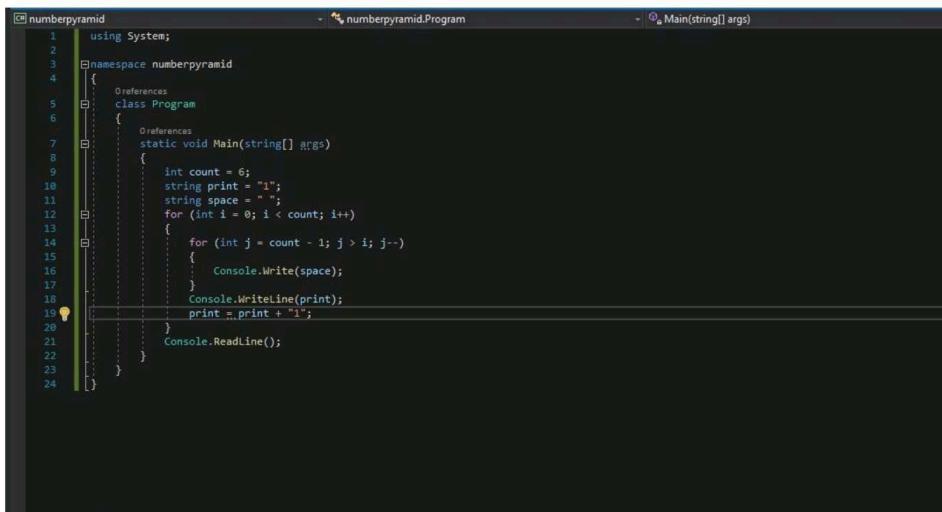
Output:



```
C:\Users\SANJAY LADDHA\source\repos\halfpyramid\halfpyramid\bin\Debug\netcoreapp3.1\halfpyramid.exe
*****
****
```

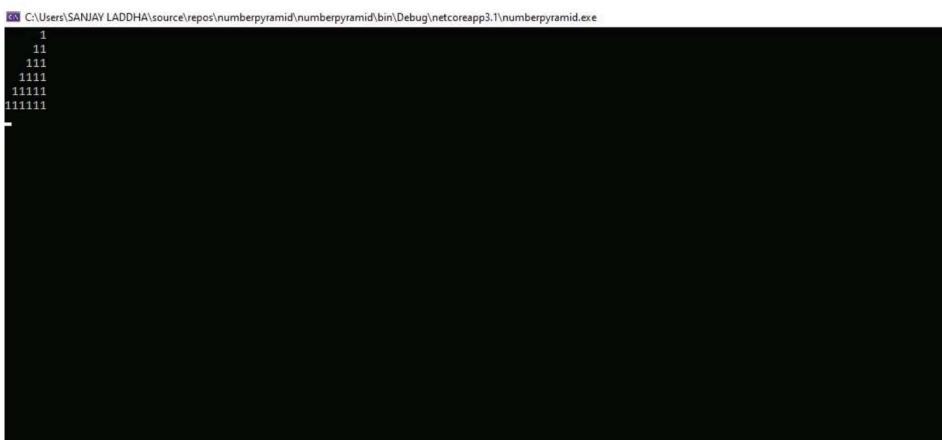
7] Write a programme to print the half pyramid pattern with numbers:

Program:-



```
numberpyramid
1  using System;
2
3  namespace numberpyramid
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              int count = 6;
10             string print = "1";
11             string space = " ";
12             for (int i = 0; i < count; i++)
13             {
14                 for (int j = count - 1; j > i; j--)
15                 {
16                     Console.Write(space);
17                 }
18                 Console.WriteLine(print);
19                 print = print + "1";
20             }
21             Console.ReadLine();
22         }
23     }
24 }
```

Output:



```
C:\Users\SANJAY LADDHA\source\repos\numberpyramid\numberpyramid\bin\Debug\netcoreapp3.1\numberpyramid.exe
1
11
111
1111
11111
```

8] Write a programme to print half pyramid inverse pattern:

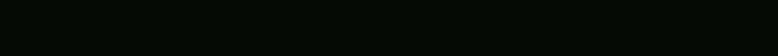
Program:-

```
invertedhalfpyramid.cs  C:\invertedhalfpyramid\bin\Debug\invertedhalfpyramid.exe  Main(string[] args)

1  using System;
2
3  namespace invertedhalfpyramid
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              for (int row = 8; row >= 1; --row)
10             {
11                 for (int col = 1; col <= row; ++col)
12                 {
13                     Console.WriteLine("*");
14                 }
15
16                 Console.WriteLine();
17             }
18             Console.ReadLine();
19         }
20     }
21 }
```

Output:

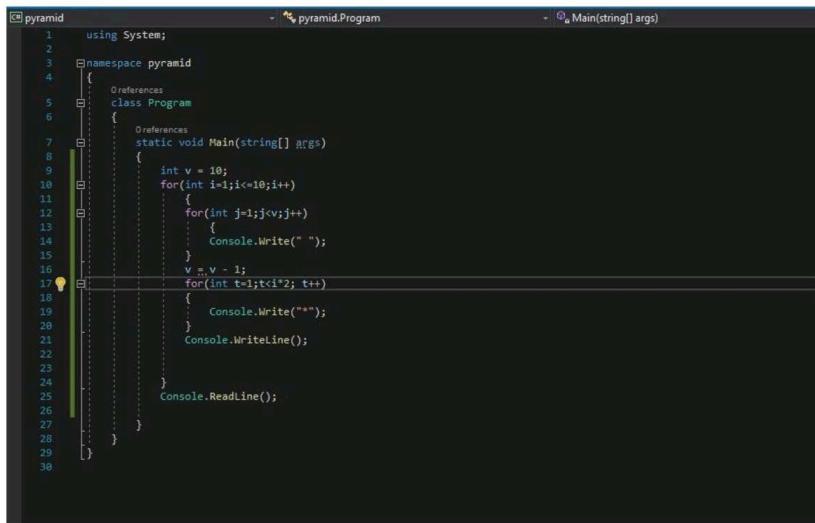
```
C:\Users\SANJAY LADDHA\source\repos\invertedhalfpyramid\invertedhalfpyramid\bin\Debug\netcoreapp3.1\invertedhalfpyramid.exe
```



```
*
**
***
****
*****
******
*****
```

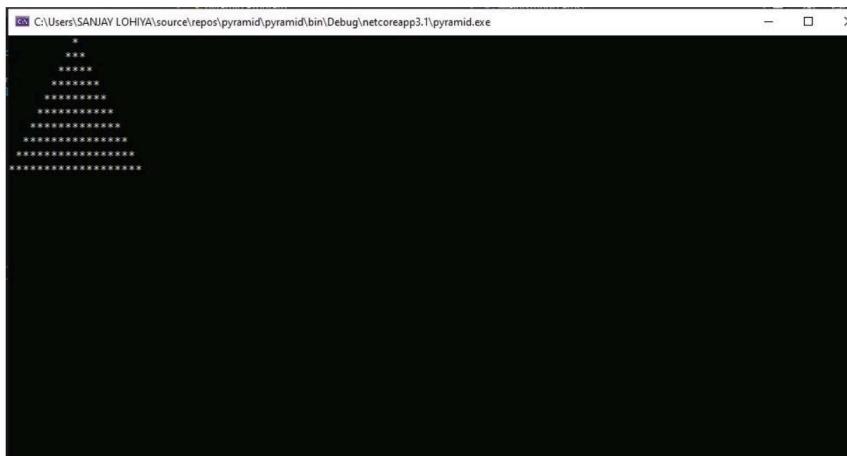
9] write a programme to print the pyramid pattern:

Program:-



```
pyramid          pyramid.Program      Main(string[] args)
1   using System;
2
3   namespace pyramid
4   {
5       class Program
6       {
7           static void Main(string[] args)
8           {
9               int v = 10;
10              for(int i=1;i<=10;i++)
11              {
12                  for(int j=1;j<v;j++)
13                  {
14                      Console.Write(" ");
15                  }
16                  v--;
17                  for(int t=1;t<i*2; t++)
18                  {
19                      Console.Write("*");
20                  }
21                  Console.WriteLine();
22
23
24
25
26             }
27         }
28     }
29 }
```

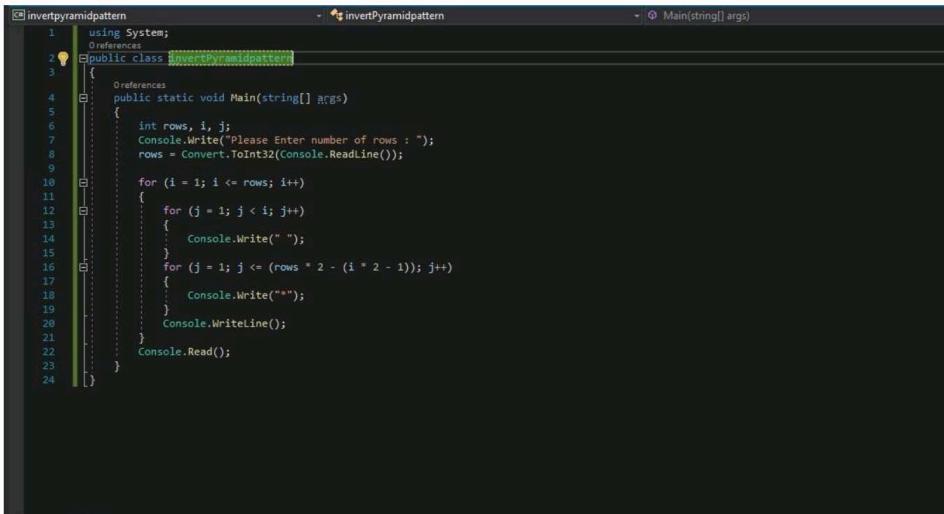
Output:



```
C:\Users\SANJAY LOHIYA\source\repos\pyramid\pyramid\bin\Debug\netcoreapp3.1\pyramid.exe
*
**
***
****
*****
******
*****
****
 ***
 **
 *
```

10] Write a programme to print the inverse pyramid pattern:

Program:-



```
1  using System;
2  References
3  {
4      References
5      public static void Main(string[] args)
6      {
7          int rows, i, j;
8          Console.Write("Please Enter number of rows : ");
9          rows = Convert.ToInt32(Console.ReadLine());
10         for (i = 1; i <= rows; i++)
11         {
12             for (j = 1; j < i; j++)
13             {
14                 Console.Write(" ");
15             }
16             for (j = 1; j <= (rows * 2 - (i * 2 - 1)); j++)
17             {
18                 Console.Write("*");
19             }
20             Console.WriteLine();
21         }
22     }
23 }
24 }
```

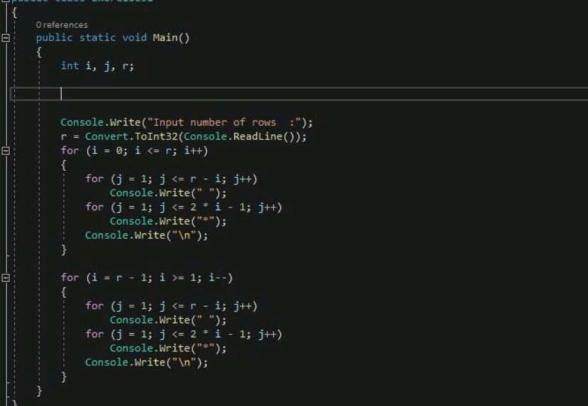
Output:



```
C:\Users\SANJAY LADDHA\source\repos\invertpyramidpattern\invertpyramidpattern\bin\Debug\netcoreapp3.1\invertpyramidpattern.exe
Please Enter number of rows : 6
*****
*****
****
 ***
 *

```

11] Write a programme to print the diamond pattern:
Program:-



The screenshot shows a code editor with a dark theme. The code is a C# program named `Exercise31` located in a file named `Main()`. The code prints a diamond pattern to the console. It uses nested loops to determine the number of spaces and asterisks to print at each row. A yellow lightbulb icon is visible on the left margin, indicating a potential issue or suggestion.

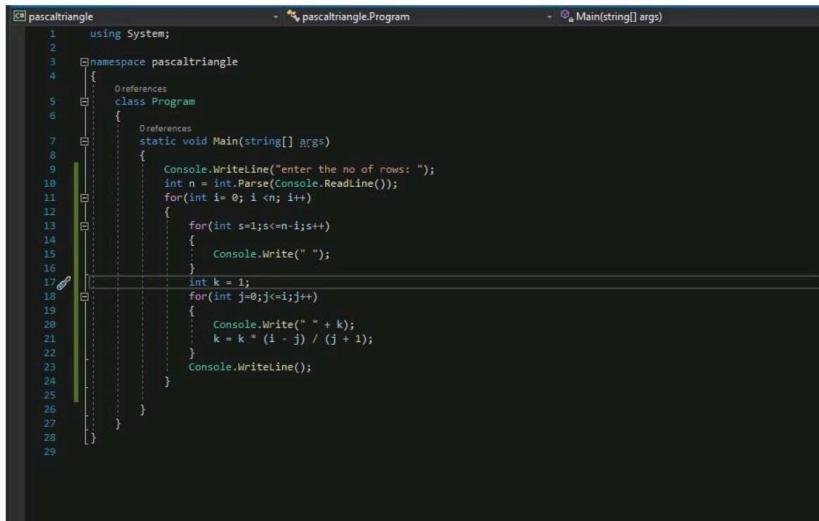
```
1  using System;
2  References
3  public class Exercise31
4  {
5      References
6      public static void Main()
7      {
8          References
9          int i, j, r;
10         |
11         Console.WriteLine("Input number of rows :");
12         r = Convert.ToInt32(Console.ReadLine());
13         for (i = 0; i <= r; i++)
14         {
15             for (j = 1; j <= r - i; j++)
16                 Console.Write(" ");
17             for (j = 1; j <= 2 * i - 1; j++)
18                 Console.Write("*");
19             Console.WriteLine("\n");
20         }
21         for (i = r - 1; i >= 1; i--)
22         {
23             for (j = 1; j <= r - i; j++)
24                 Console.Write(" ");
25             for (j = 1; j <= 2 * i - 1; j++)
26                 Console.Write("*");
27             Console.WriteLine("\n");
28         }
29     }
30 }
```

Output:

```
Microsoft Visual Studio Debug Console
Input number of rows :8
      *
     ***
    *****
   ******
  *****
 *****
*****
 *****
  ****
   ***
    *
 *
```

12] Write a programme to print the pascal's triangle:

Program:-



```
pascaltriangle
1  using System;
2
3  namespace pascaltriangle
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("enter the no of rows: ");
10             int n = int.Parse(Console.ReadLine());
11             for(int i= 0; i <n; i++)
12             {
13                 for(int s=1;s<=n-i;s++)
14                 {
15                     Console.Write(" ");
16                 }
17                 int k = 1;
18                 for(int j=0;j<=i;j++)
19                 {
20                     Console.Write(" " + k);
21                     k = k * (i - j) / (j + 1);
22                 }
23                 Console.WriteLine();
24             }
25         }
26     }
27 }
28 }
```

Output:



```
Microsoft Visual Studio Debug Console
enter the no of rows:
3
1
1 1
1 2 1
```

13] Write a programme to compare 2 string without using string library function:

Program:-

```
strings          Exercise6           Main()
1  using System;
2  public class Exercise6
3  {
4      public static void Main()
5      {
6          string str1, str2;
7          int flg = 0;
8          int l1 = 0, l2, yn = 0;
9          Console.WriteLine("Input the 1st string : ");
10         str1 = Console.ReadLine();
11         Console.WriteLine("Input the 2nd string : ");
12         str2 = Console.ReadLine();
13         l1 = str1.Length;
14         l2 = str2.Length;
15
16         if (l1 == l2)
17         {
18             for (i = 0; i < l1; i++)
19             {
20                 if (str1[i] != str2[i])
21                 {
22                     yn = 1;
23                     i = l1;
24                 }
25             }
26         }
27
28         if (l1 == l2)
29             flg = 0;
30         else if (l1 > l2)
31             flg = 1;
32         else if (l1 < l2)
33             flg = -1;
34
35 }
```

```
strings          Exercise6           Main()
35
36         if (flg == 0)
37         {
38             if (yn == 0)
39                 Console.WriteLine("\nThe length of both strings are equal and \nalso, both strings are same.\n\n");
40             else
41                 Console.WriteLine("\nThe length of both strings are equal \nbut they are not same.\n\n");
42         }
43         else if (flg == -1)
44         {
45             Console.WriteLine("\nThe length of the first string is smaller than second.\n\n");
46         }
47         else
48         {
49             Console.WriteLine("\nThe length of the first string is greater than second.\n\n");
50         }
51     }
52 }
53 ]
```

Output:

```
Microsoft Visual Studio Debug Console
Input the 1st string : sajal
Input the 2nd string : rupali

The length of the first string is smaller than second.

C:\Users\SANJAY LADDHA\source\repos\strings\strings\bin\Debug\netcoreapp3.1\strings.exe (process 10440) exited with code 0.
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 . . .
```

14] Write a programme to count a total numbers of alphabets , digits and special characters in a string:

Program:-

```
counts    Exercise7    Main()
```

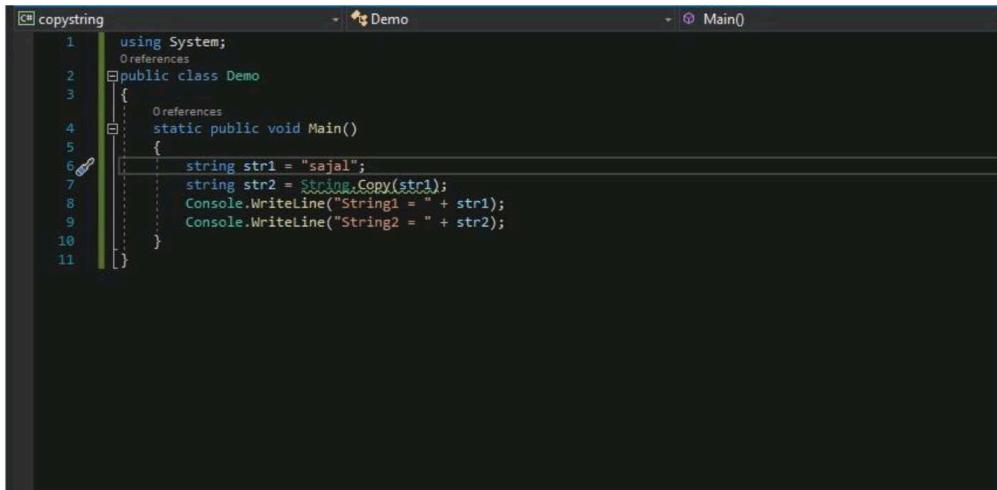
```
1  using System;
2  References
3  public class Exercise7
4  {
5      References
6      public static void Main()
7      {
8          string str;
9          int alp, digit, splch, i, l;
10         alp = digit = splch = i = 0;
11         Console.WriteLine("Input the string : ");
12         str = Console.ReadLine();
13         l = str.Length;
14         while (i < l)
15         {
16             if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z'))
17             {
18                 alp++;
19             }
20             else if (str[i] >= '0' && str[i] <= '9')
21             {
22                 digit++;
23             }
24             else
25             {
26                 splch++;
27             }
28             i++;
29         }
30         Console.WriteLine("Number of Alphabets in the string is : {0}\n", alp);
31         Console.WriteLine("Number of Digits in the string is : {0}\n", digit);
32         Console.WriteLine("Number of Special characters in the string is : {0}\n\n", splch);
33     }
34 }
```

Output:

```
Microsoft Visual Studio Debug Console
Input the string : sayjal@123
Number of Alphabets in the string is : 5
Number of Digits in the string is : 3
Number of Special characters in the string is : 1

C:\Users\SANJAY LADDHA\source\repos\counts\counts\bin\Debug\netcoreapp3.1\counts.exe (process 13604) exited with code 0.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 . . .
```

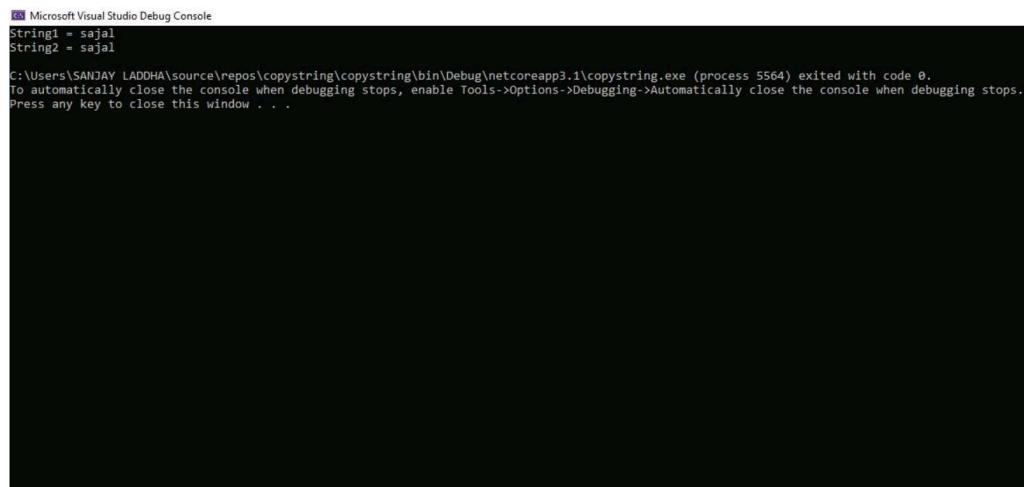
15] Write a programme to copy one string to another string: Program:-



A screenshot of the Microsoft Visual Studio IDE. The code editor window is open with the file name 'copystring.cs'. The code is as follows:

```
1  using System;
2  References
3  public class Demo
4  {
5      References
6      static public void Main()
7      {
8          string str1 = "sajal";
9          string str2 = String.Copy(str1);
10         Console.WriteLine("String1 = " + str1);
11         Console.WriteLine("String2 = " + str2);
12     }
13 }
```

Output:

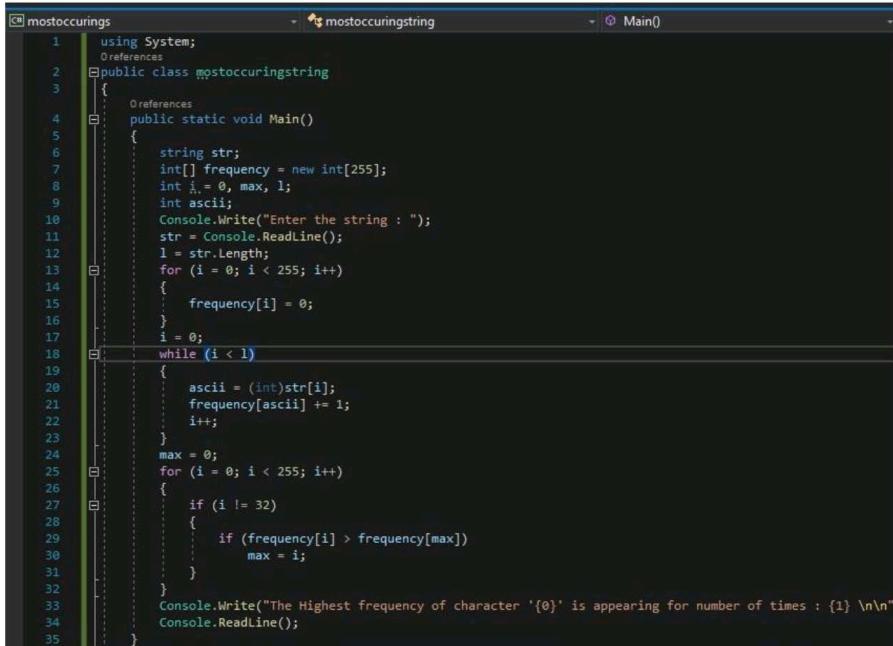


A screenshot of the Microsoft Visual Studio Debug Console window. The output is:

```
Microsoft Visual Studio Debug Console
String1 = sajal
String2 = sajal
C:\Users\SANJAY LADDHA\source\repos\copystring\copystring\bin\Debug\netcoreapp3.1\copystring.exe (process 5564) exited with code 0.
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 . . .
```

16] Write a programme to find maximum occurring characterin a string:

Program:-



```
1  using System;
2  References:
3  public class mostoccuringstring
4  {
5      References:
6      public static void Main()
7      {
8          string str;
9          int[] frequency = new int[255];
10         int i = 0, max, l;
11         int ascii;
12         Console.WriteLine("Enter the string : ");
13         str = Console.ReadLine();
14         l = str.Length;
15         for (i = 0; i < 255; i++)
16         {
17             frequency[i] = 0;
18         }
19         i = 0;
20         while (i < l)
21         {
22             ascii = (int)str[i];
23             frequency[ascii] += 1;
24             i++;
25         }
26         max = 0;
27         for (i = 0; i < 255; i++)
28         {
29             if (i != 32)
30             {
31                 if (frequency[i] > frequency[max])
32                     max = i;
33             }
34         }
35         Console.WriteLine("The Highest frequency of character '{0}' is appearing for number of times : {1} \n\n");
36     }
37 }
```

Output:

```
C:\Users\SANJAY LADDHA\source\repos\mostoccuring\mostoccuring\bin\Debug\netcoreapp3.1\mostoccuring.exe
Enter the string : sajal laddha
The Highest frequency of character 'a' is appearing for number of times : 4
```

17] Write a programme to check whether a given substring is present in the given string:

Program:-

The screenshot shows the Microsoft Visual Studio IDE interface. The title bar says "checksubstring". The code editor contains the following C# code:

```
1  using System;
2  0 references
3  public class checksubstring
4  {
5      0 references
6      public static void Main()
7      {
8          string str1, str2;
9          bool m;
10
11         Console.Write("Input the string : ");
12         str1 = Console.ReadLine();
13
14         Console.Write("Input the substring to search : ");
15         str2 = Console.ReadLine();
16         m = str1.Contains(str2); // boolean value tapped here
17         if (m) // check boolean value is true or false.
18             Console.WriteLine("The substring exists in the string.\n\n");
19         else
20             Console.WriteLine("The substring is not exists in the string. \n\n");
21     }
}
```

Output:

(substring exists

)

The screenshot shows the Microsoft Visual Studio Debug Console window. It displays the following output:

```
Microsoft Visual Studio Debug Console
Input the string : sajal sanjay laddha
Input the substring to search : sanjay
The substring exists in the string.

C:\Users\SANJAY LADDHA\source/repos\checksubstring\checksubstring\bin\Debug\netcoreapp3.1\checksubstring.exe (process 13264) exited with code 0.
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 . . .
```

(substring doesn't exists)

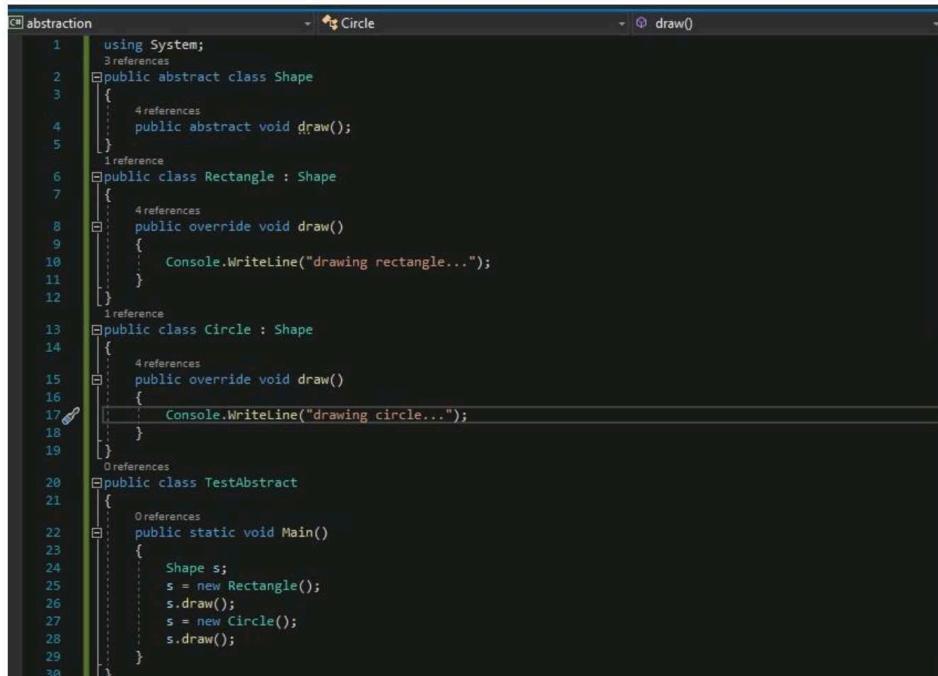
The screenshot shows the Microsoft Visual Studio Debug Console window. It displays the following output:

```
Microsoft Visual Studio Debug Console
Input the string : sajal laddha
Input the substring to search : sanjaY
The substring is not exists in the string.

C:\Users\SANJAY LADDHA\source/repos\checksubstring\checksubstring\bin\Debug\netcoreapp3.1\checksubstring.exe (process 9848) exited with code 0.
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 . . .
```

18] Write a programme for abstraction:

Program:-

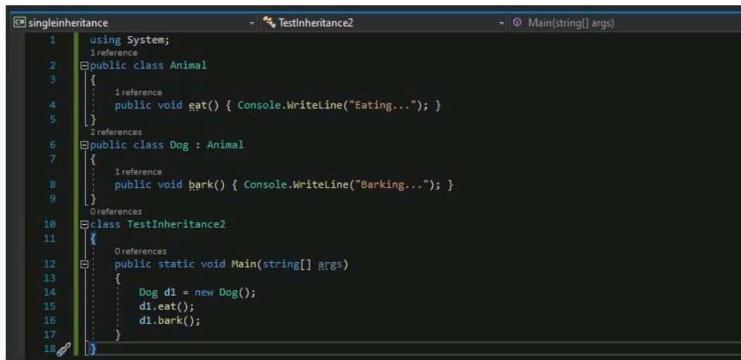


```
abstraction
1  using System;
2  3 references
3  public abstract class Shape
4  {
5      4 references
6      public abstract void draw();
7  }
8  1 reference
9  public class Rectangle : Shape
10 {
11     4 references
12     public override void draw()
13     {
14         Console.WriteLine("drawing rectangle...");
15     }
16 }
17 1 reference
18 public class Circle : Shape
19 {
20     4 references
21     public override void draw()
22     {
23         Console.WriteLine("drawing circle...");
24     }
25 }
26 0 references
27 public class TestAbstract
28 {
29     0 references
30     public static void Main()
31     {
32         Shape s;
33         s = new Rectangle();
34         s.draw();
35         s = new Circle();
36         s.draw();
37     }
38 }
```

Output:

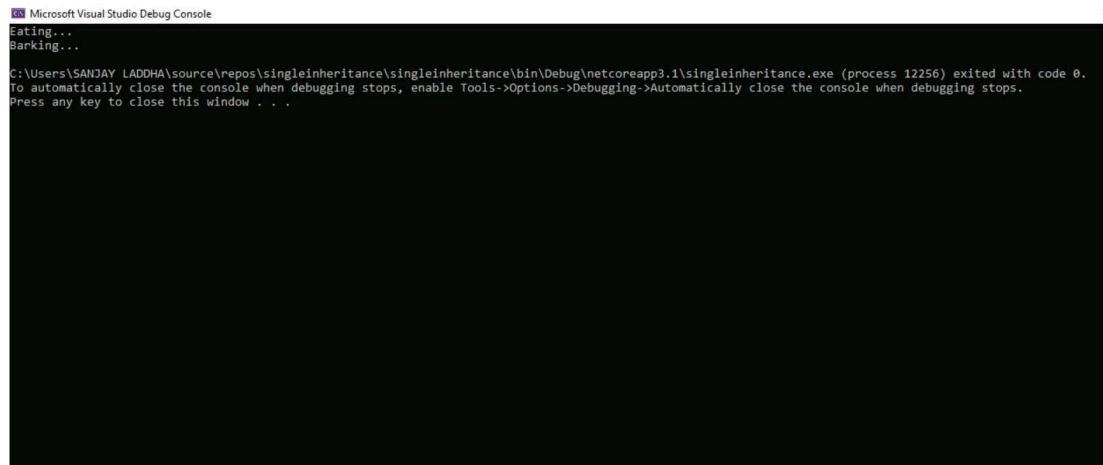
```
Microsoft Visual Studio Debug Console
drawing rectangle...
drawing circle...
C:\Users\SANDAY LADDHA\source\repos\abstraction\abstraction\bin\Debug\netcoreapp3.1\abstraction.exe (process 11188) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

19] Write a programme for single inheritance:



```
singleinheritance      TestInheritance2      Main(string[] args)
1  using System;
2  reference
3  public class Animal
4  {
5      reference
6      public void eat() { Console.WriteLine("Eating..."); }
7  }
8  references
9  public class Dog : Animal
10 {
11     reference
12     public void bark() { Console.WriteLine("Barking..."); }
13 }
14 References
15 class TestInheritance2
16 {
17     References
18     public static void Main(string[] args)
19     {
20         Dog d1 = new Dog();
21         d1.eat();
22         d1.bark();
23     }
24 }
```

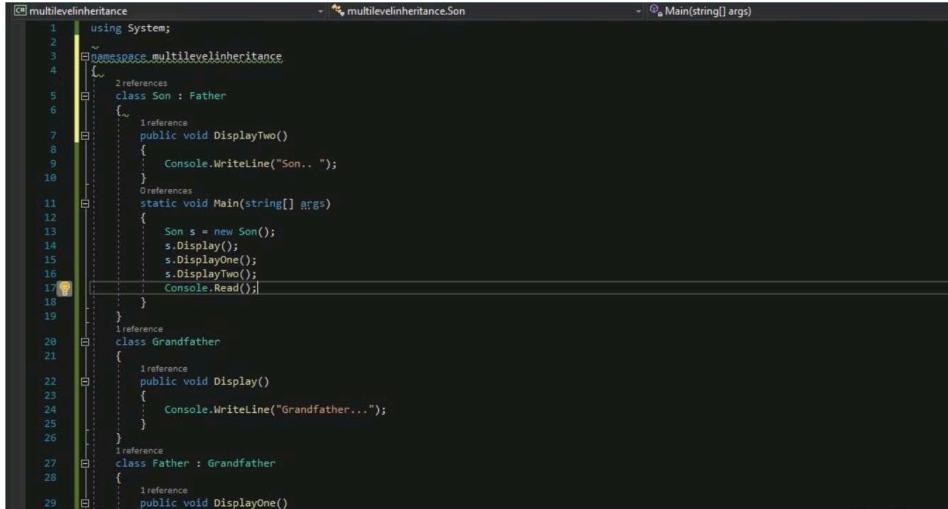
Output:



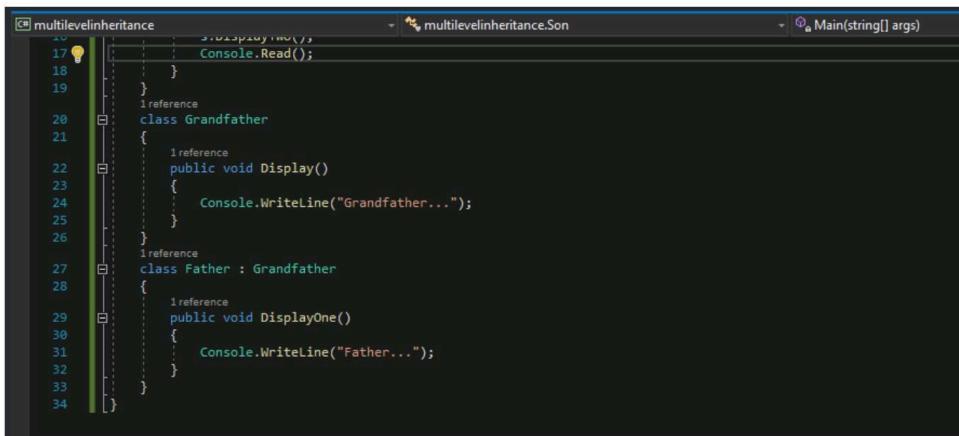
```
Microsoft Visual Studio Debug Console
Eating...
Barking...
C:\Users\SANJAY LADDHA\source\repos\singleinheritance\bin\Debug\netcoreapp3.1\singleinheritance.exe (process 12256) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

20] Write a programme for multi level inheritance:

Program:-



```
1 using System;
2 ...
3 namespace multilevelinheritance
4 {
5     //reference
6     class Son : Father
7     {
8         //reference
9         public void DisplayTwo()
10        {
11            Console.WriteLine("Son.. ");
12        }
13        //reference
14        static void Main(string[] args)
15        {
16            Son s = new Son();
17            s.Display();
18            s.DisplayOne();
19            s.DisplayTwo();
20            Console.Read();
21        }
22     }
23     //reference
24     class Grandfather
25     {
26         //reference
27         public void Display()
28         {
29             Console.WriteLine("Grandfather...");
30         }
31     }
32     //reference
33     class Father : Grandfather
34     {
35         //reference
36         public void DisplayOne()
37         {
38             Console.WriteLine("Father...");
39         }
40     }
41 }
```



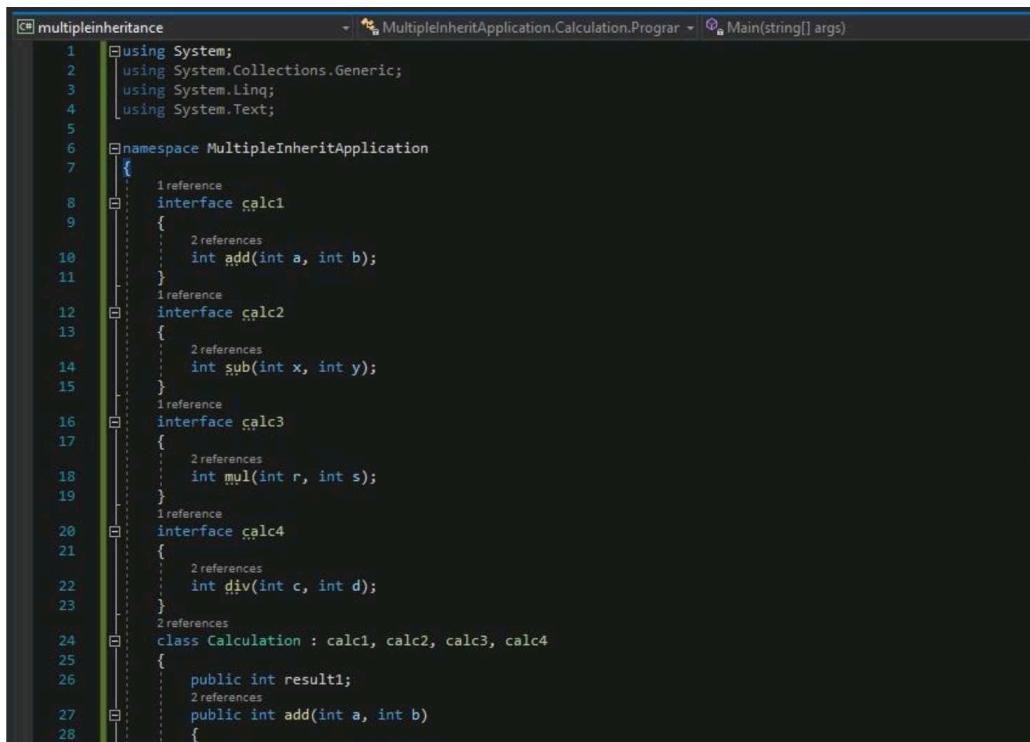
```
10    s.DisplayTwo();
11    Console.Read();
12 }
13
14 //reference
15 class Grandfather
16 {
17     //reference
18     public void Display()
19     {
20         Console.WriteLine("Grandfather...");
21     }
22 }
23
24 //reference
25 class Father : Grandfather
26 {
27     //reference
28     public void DisplayOne()
29     {
30         Console.WriteLine("Father...");
31     }
32 }
33
34 }
```

Output:

```
C:\Users\SANJAY LADDHA\source\repos\multilevelinheritance\multilevelinheritance\bin\Debug\netcoreapp3.1\multilevelinheritance.exe
Grandfather...
Father...
Son..
```

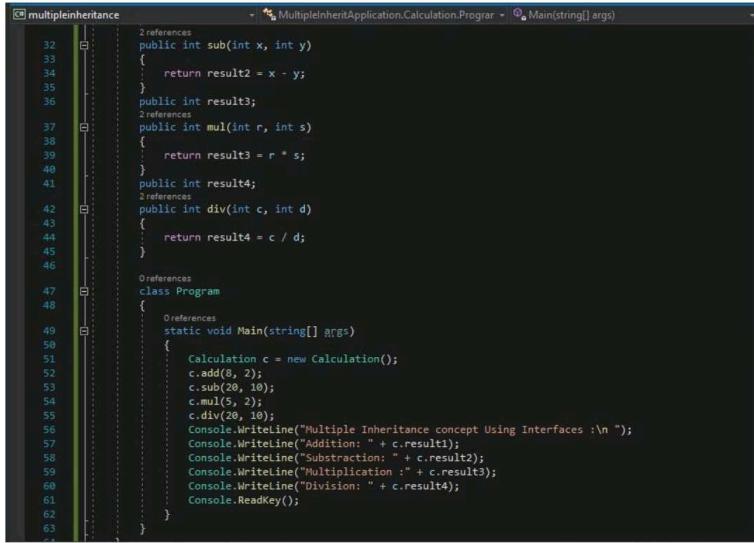
21] Write a programme for multiple inheritance: (Multiple inheritance runs with the help of interface not class)

Program:-



The screenshot shows a code editor window with the following C# code:

```
multipleinheritance
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5
6  namespace MultipleInheritApplication
7  {
8      interface calc1
9      {
10         int add(int a, int b);
11     }
12     interface calc2
13     {
14         int sub(int x, int y);
15     }
16     interface calc3
17     {
18         int mul(int r, int s);
19     }
20     interface calc4
21     {
22         int div(int c, int d);
23     }
24     class Calculation : calc1, calc2, calc3, calc4
25     {
26         public int result1;
27         public int add(int a, int b)
28     }
```



The screenshot shows a Microsoft Visual Studio code editor window titled "multipleinheritance". The code is written in C# and defines a class "Program" that implements four interfaces: Calculation, IAddition, ISubtraction, IMultiplication, and IDivision. The class contains a static Main method that creates an instance of Calculation and performs various arithmetic operations using the interface methods.

```
32     public int sub(int x, int y)
33     {
34         return result2 = x - y;
35     }
36     public int result3;
37     public int mul(int r, int s)
38     {
39         return result3 = r * s;
40     }
41     public int result4;
42     public int div(int c, int d)
43     {
44         return result4 = c / d;
45     }
46
47     class Program
48     {
49         static void Main(string[] args)
50         {
51             Calculation c = new Calculation();
52             c.add(8, 2);
53             c.sub(20, 10);
54             c.mul(5, 2);
55             c.div(20, 10);
56             Console.WriteLine("Multiple Inheritance concept Using Interfaces :\\n ");
57             Console.WriteLine("Addition: " + c.result1);
58             Console.WriteLine("Subtraction: " + c.result2);
59             Console.WriteLine("Multiplication :" + c.result3);
60             Console.WriteLine("Division: " + c.result4);
61             Console.ReadKey();
62         }
63     }
}
```

Output

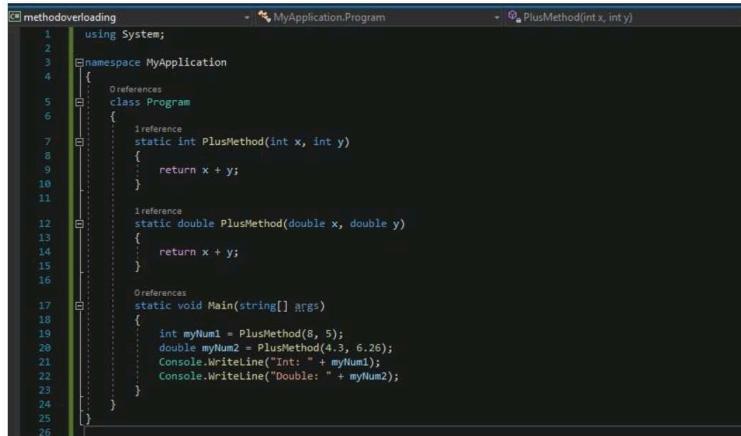


The screenshot shows the Microsoft Visual Studio Debug Console window. The output displays the results of the arithmetic operations performed by the "Program" class. The console shows the addition, subtraction, multiplication, and division results, all of which are 10. The console window also includes standard debugging information such as the current working directory and the exit code.

```
Microsoft Visual Studio Debug Console
Multiple Inheritance concept Using Interfaces :
Addition: 10
Subtraction: 10
Multiplication :10
Division: 2
C:\Users\SANDAY LADDHA\source\repos\multipleinheritance\multipleinheritance\bin\Debug\netcoreapp3.1\multipleinheritance.exe (process 796) exited with code 0.
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 . . .
```

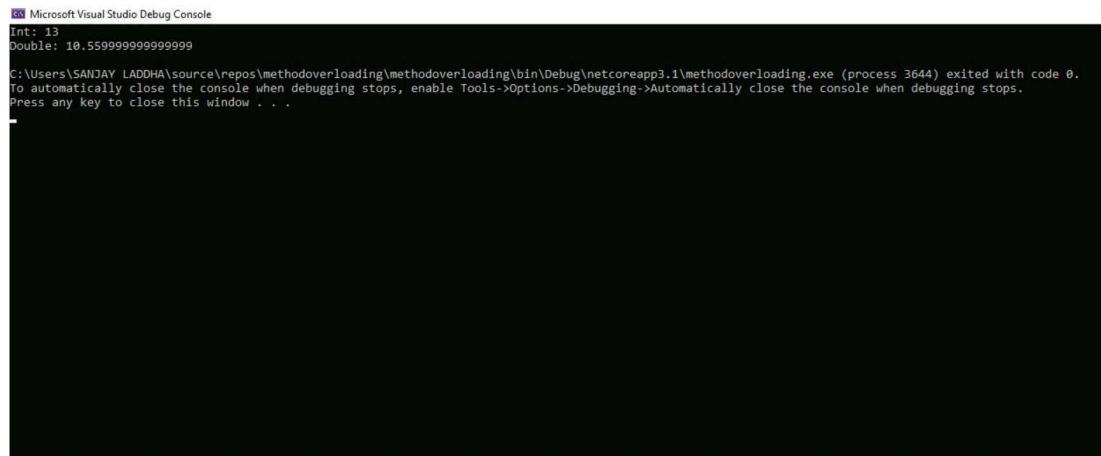
22] Write a programme for method overloading:

Program:-



```
methodoverloading.cs 1  using System;
2
3  namespace MyApplication
4  {
5      References
6      class Program
7      {
8          References
9          static int PlusMethod(int x, int y)
10         {
11             ...
12             return x + y;
13         }
14
15         References
16         static double PlusMethod(double x, double y)
17         {
18             ...
19             return x + y;
20         }
21
22         References
23         static void Main(string[] args)
24         {
25             int myNum1 = PlusMethod(8, 5);
26             double myNum2 = PlusMethod(4.5, 6.26);
27             Console.WriteLine("Int: " + myNum1);
28             Console.WriteLine("Double: " + myNum2);
29         }
30     }
31 }
```

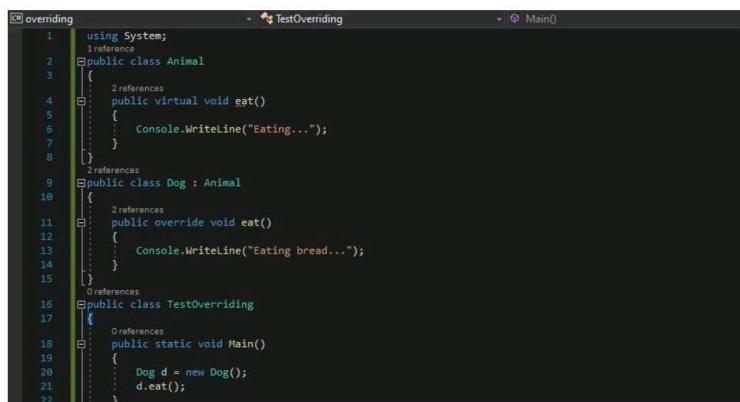
Output:



```
Microsoft Visual Studio Debug Console
Int: 13
Double: 10.55999999999999
C:\Users\SANJAY LADDHA\source\repos\methodoverloading\methodoverloading\bin\Debug\netcoreapp3.1\methodoverloading.exe (process 3644) exited with code 0.
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 . . .
```

23] Write a programme for method overriding:

Program:-



```
override.cs  TestOverriding  Main()
```

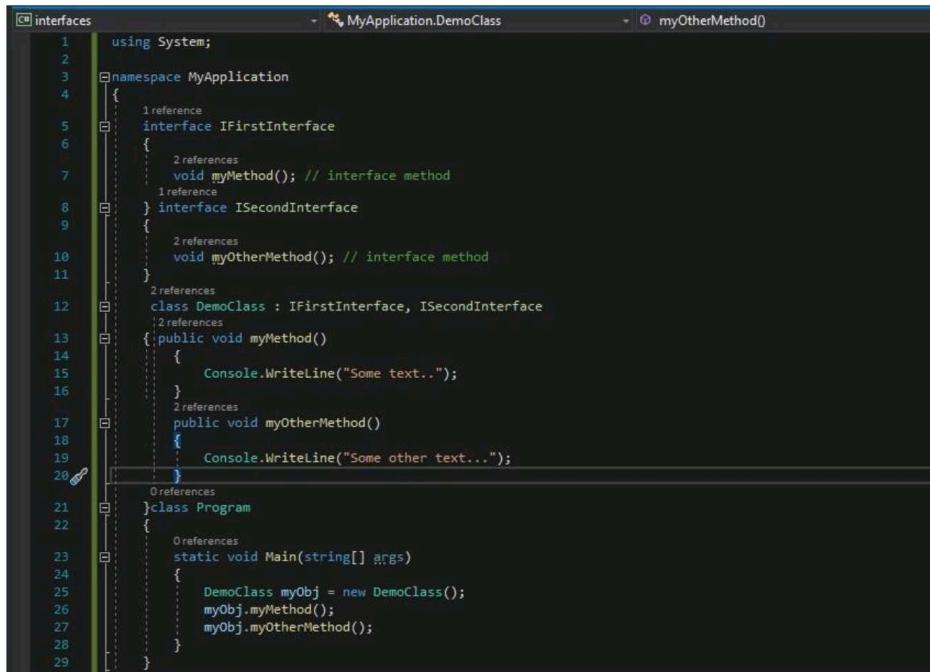
```
1  using System;
2  reference
3  public class Animal
4  {
5      2 references
6      public virtual void eat()
7      {
8          Console.WriteLine("Eating...");
9      }
10 }
11 2 references
12 public class Dog : Animal
13 {
14     2 references
15     public override void eat()
16     {
17         Console.WriteLine("Eating bread...");
18     }
19 }
20 0 references
21 public class TestOverriding
22 {
23     0 references
24     public static void Main()
25     {
26         Dog d = new Dog();
27         d.eat();
28     }
29 }
```

Output:



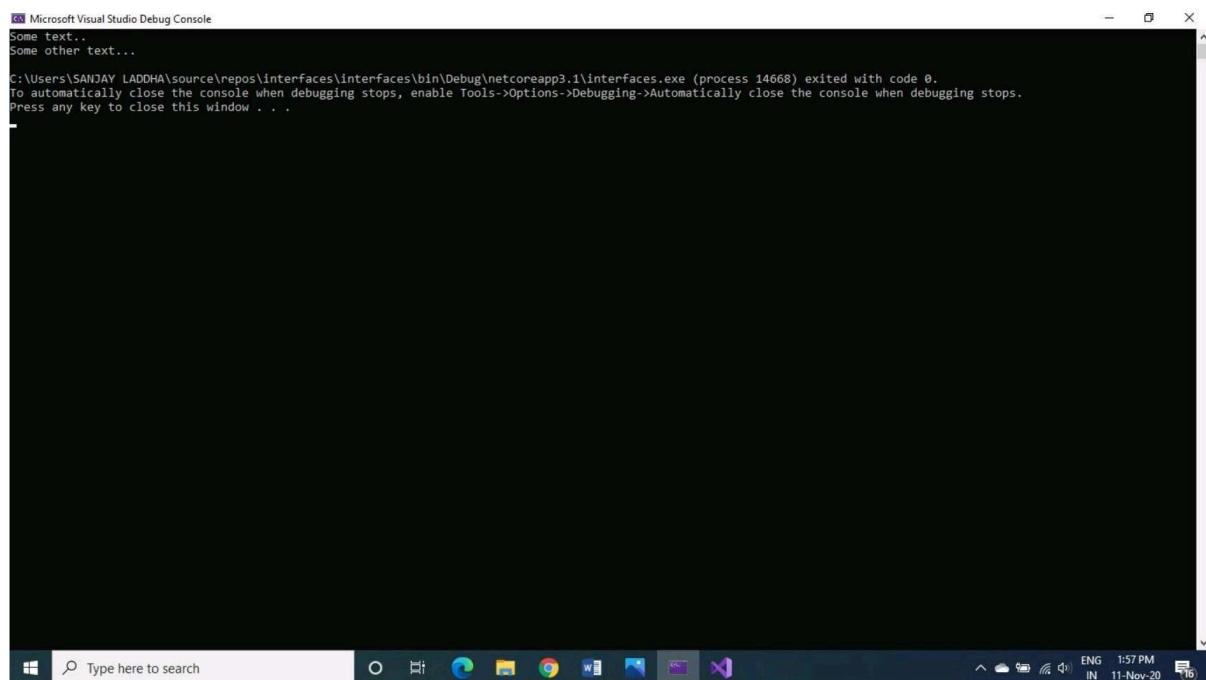
```
Microsoft Visual Studio Debug Console
Eating bread...
C:\Users\SANJAY LADDHA\source\repos\overriding\bin\Debug\netcoreapp3.1\overriding.exe (process 7204) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

24] Write a programme for interface: Program:-



```
interfaces
1  using System;
2
3  namespace MyApplication
4  {
5      interface IFirstInterface
6      {
7          void myMethod(); // interface method
8      }
9      interface ISecondInterface
10     {
11         void myOtherMethod(); // interface method
12     }
13     class DemoClass : IFirstInterface, ISecondInterface
14     {
15         public void myMethod()
16         {
17             Console.WriteLine("Some text..");
18         }
19         public void myOtherMethod()
20         {
21             Console.WriteLine("Some other text..");
22         }
23     }
24     class Program
25     {
26         static void Main(string[] args)
27         {
28             DemoClass myObj = new DemoClass();
29             myObj.myMethod();
30             myObj.myOtherMethod();
31         }
32     }
33 }
```

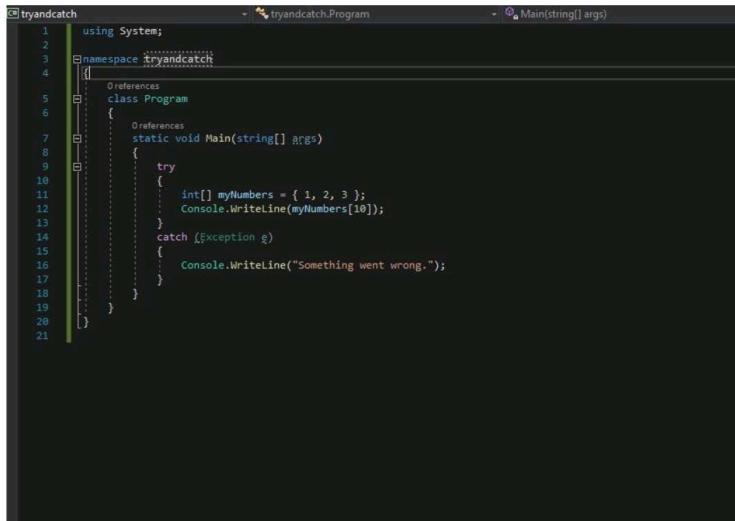
Output:



```
Microsoft Visual Studio Debug Console
Some text..
Some other text...
C:\Users\SANJAY LADDHA\source\repos\interfaces\interfaces\bin\Debug\netcoreapp3.1\interfaces.exe (process 14668) exited with code 0.
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 . . .
```

25] Write a programme for exception handling through try and catch:

Program:-



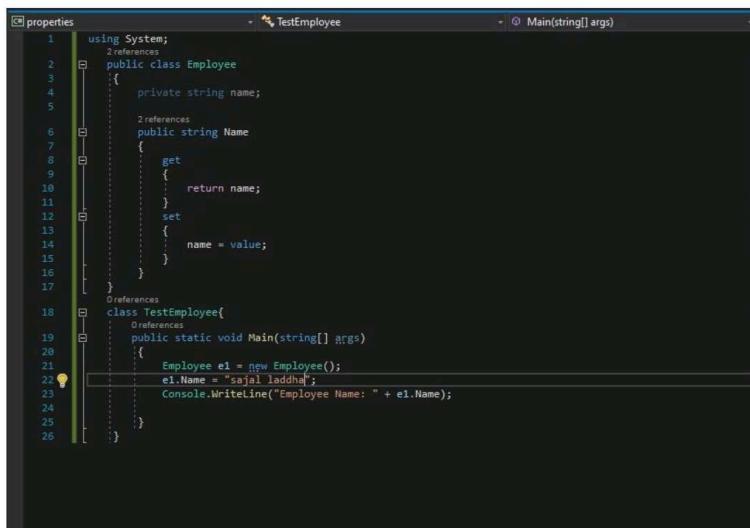
```
tryandcatch          tryandcatch.Program        Main(string[] args)
1  using System;
2
3  namespace tryandcatch
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              try
10             {
11                 int[] myNumbers = { 1, 2, 3 };
12                 Console.WriteLine(myNumbers[10]);
13             }
14             catch (Exception e)
15             {
16                 Console.WriteLine("Something went wrong.");
17             }
18         }
19     }
20 }
21
```

Output:

```
Something went wrong.
C:\Users\SANJAY LADDHA\source\repos\tryandcatch\tryandcatch\bin\Debug\netcoreapp3.1\tryandcatch.exe (process 9680) exited with code 0.
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 . . .
```

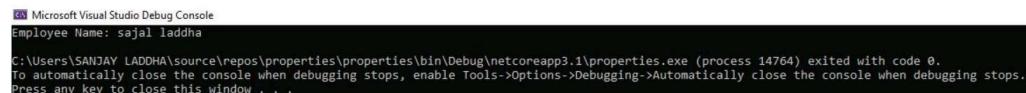
26] Write a programme for properties.

Program:-



```
1  using System;
2  references
3  public class Employee
4  {
5      private string name;
6
7      references
8      public string Name
9      {
10         get
11         {
12             return name;
13         }
14         set
15         {
16             name = value;
17         }
18     }
19     References
20     class TestEmployee{
21         References
22         public static void Main(string[] args)
23         {
24             Employee e1 = new Employee();
25             e1.Name = "sajal laddha";
26             Console.WriteLine("Employee Name: " + e1.Name);
27         }
28     }
29 }
```

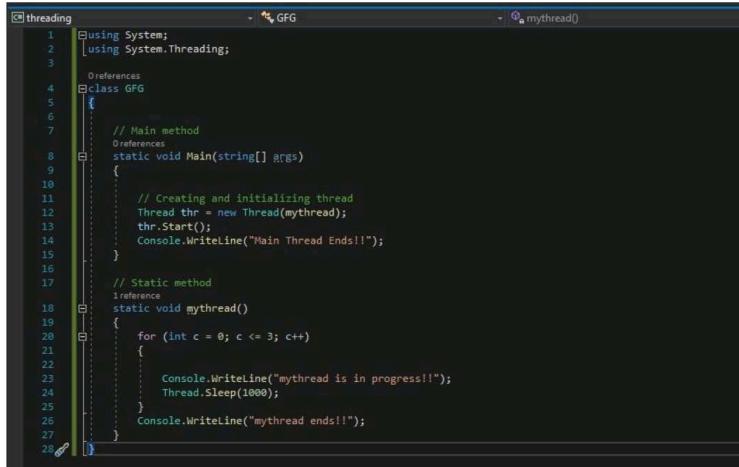
Output:



```
Microsoft Visual Studio Debug Console
Employee Name: sajal laddha
C:\Users\SANJAY LADDHA\source\repos\properties\properties\bin\Debug\netcoreapp3.1\properties.exe (process 14764) exited with code 0.
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 . . .
```

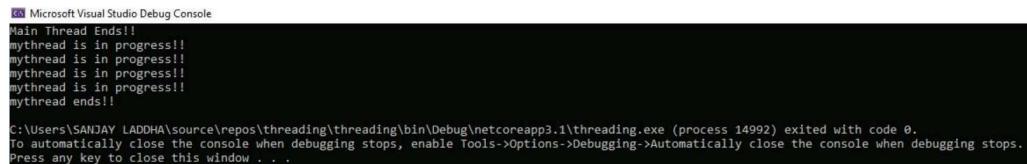
27] Write a programme for threading.

Program:-



```
threading
1  using System;
2  [using System.Threading;
3
4  References
5  class GFG
6  {
7
8      // Main method
9      static void Main(string[] args)
10     {
11
12         // Creating and initializing thread
13         Thread thr = new Thread(mythread);
14         thr.Start();
15         Console.WriteLine("Main Thread Ends!!");
16     }
17
18     // Static method
19     static void mythread()
20     {
21
22         for (int c = 0; c <= 3; c++)
23         {
24
25             Console.WriteLine("mythread is in progress!!");
26             Thread.Sleep(1000);
27         }
28         Console.WriteLine("mythread ends!!");
29     }
30 }
```

Output:

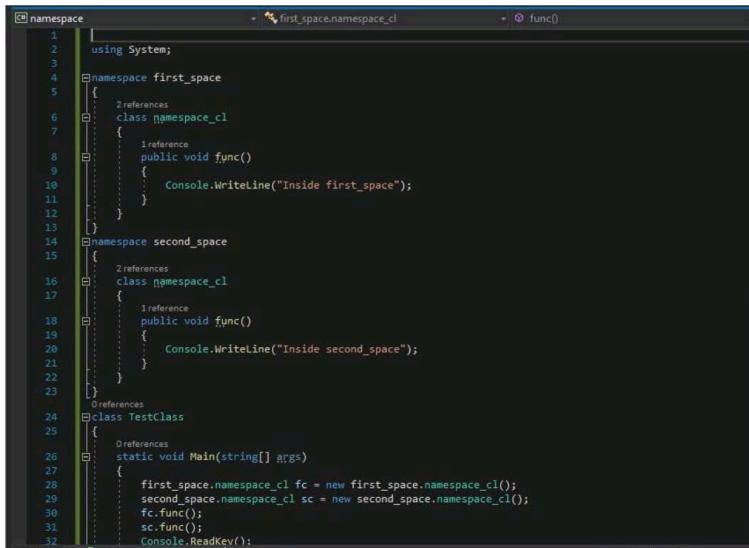


```
Microsoft Visual Studio Debug Console
Main Thread Ends!!
mythread is in progress!!
mythread is in progress!!
mythread is in progress!!
mythread is in progress!!
mythread ends!!

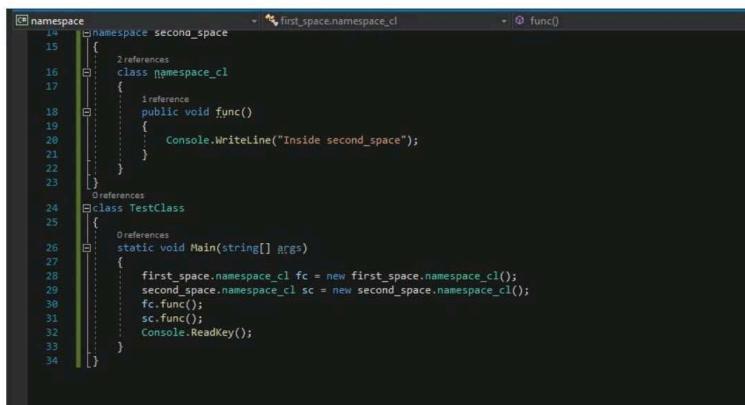
C:\Users\SANJAY LADDAH\source\repos\threading\threading\bin\Debug\netcoreapp3.1\threading.exe (process 14992) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

29] Write a programme using namespace.

Program:-



```
namespace
1  using System;
2
3  namespace first_space
4  {
5      2 references
6      class namespace_cl
7      {
8          1 reference
9          public void func()
10         {
11             Console.WriteLine("Inside first_space");
12         }
13     }
14
15     namespace second_space
16     {
17         2 references
18         class namespace_cl
19         {
20             1 reference
21             public void func()
22             {
23                 Console.WriteLine("Inside second_space");
24             }
25         }
26     }
27 }
28
29 References
30 class TestClass
31 {
32     0 references
33     static void Main(string[] args)
34     {
35         first_space.namespace_cl fc = new first_space.namespace_cl();
36         second_space.namespace_cl sc = new second_space.namespace_cl();
37         fc.func();
38         sc.func();
39         Console.ReadKey();
40     }
41 }
```



```
namespace
14  namespace second_space
15  {
16      2 references
17      class namespace_cl
18      {
19          1 reference
20          public void func()
21          {
22              Console.WriteLine("Inside second_space");
23          }
24      }
25 }
26
27 References
28 class TestClass
29 {
30     0 references
31     static void Main(string[] args)
32     {
33         first_space.namespace_cl fc = new first_space.namespace_cl();
34         second_space.namespace_cl sc = new second_space.namespace_cl();
35         fc.func();
36         sc.func();
37         Console.ReadKey();
38     }
39 }
```

Output:

```
Microsoft Visual Studio Debug Console
Inside first_space
Inside second_space
C:\Users\SANJAY LADDHA\source\repos\namespace\namespace\bin\Debug\netcoreapp3.1\namespace.exe (process 13040) exited with code 0.
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 . . .
```

