

# DEV SANSKRITI VISHWAVIDYALAYA



SESSION 2018-2021  
PRACTICAL FILE  
ON  
“C#.NET”

**SUBMITTED TO:**

Mr. Chandrashekhar Patel  
Dept. of Computer Science

**SUBMITTED BY:**

Gaurvi Gaur  
B.C.A. (V Sem)

# INDEX

S.No.	Programs	Page No.
1.	Program to print Armstrong Number	1-2
2.	Program to print Factorial of a number	3-4
3.	Program to find GCD of two numbers	5-6
4.	Program to check out whether the number is prime or not	7-8
5.	Program to print Fabonacci Series	9-10
6.	Program to print Half Pyramid Star Pattern	11-12
7.	Program to print Half Pyramid Number Pattern	13-14
8.	Program to print Reverse Half Pyramid Star Pattern	15-16
9.	Program to print Pyramid Star Pattern	17-18
10.	Program to print Reverse Pyramid Star Pattern	19-20
11.	Program to print Diamond Star Pattern	21-22
12.	Program to print Pascal's Triangle	23-24
13.	Program to compare two strings without using string library function	25-26
14.	Program to count total number of alphabets, digits, and special characters in a string	27-28
15.	Program to copy one string to another string	29-30
16.	Program to find maximum occurring character in a string	31-32

17.	Program to check whether a given substring is present in the given string	33-34
18.	Program for Abstraction	35-36
19.	Program for Single Inheritance	37-38
20.	Program for Multilevel Inheritance	39-40
21.	Program for Multiple Inheritance	41-42
22.	Program for Method Overloading	43-44
23.	Program for Method Overriding	45-46
24.	Program for Interface	47-48
25.	Program for Exceptional Handling through try and catch	49-50
26.	Program for Properties	51-54
27.	Program for Threading	55-56
28.	Program for Namespace	57-58

## 1. Program to print Armstrong Number.

```
C# Armstrong Number Armstrong_Number.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Armstrong_Number
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             int m, sum = 0, t;
16             Console.WriteLine("Enter the Number:");
17             int n = Convert.ToInt32(Console.ReadLine());
18             t = n;
19             while (n > 0)
20             {
21                 m = n % 10;
22                 sum = sum + (m * m * m);
23                 n = n / 10;
24             }
25             if (t == sum)
26                 Console.Write("Armstrong Number.");
27             else
28                 Console.Write("Not Armstrong Number.");
29             Console.ReadLine();
30         }
31     }
32 }
```

**Output 1:**

```
Enter the Number:  
371  
Armstrong Number.
```

**Output 2:**

```
Enter the Number:  
123  
Not Armstrong Number.
```

## 2. Program to print Factorial of a number.

```
C# Factorial ▼ Factorial.Program

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Factorial
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             int facto = 1;
16             Console.WriteLine("Enter a number to find factorial:");
17             int n = Convert.ToInt32(Console.ReadLine());
18
19             for (int i = n; i >= 1; i--)
20             {
21                 facto = facto * i;
22             }
23             Console.WriteLine(facto);
24             Console.ReadLine();
25         }
26     }
27 }
28
29
```

### Output:

```
Enter a number to find factorial:  
6  
720
```

### 3. Program to find GCD of two numbers.

```
C# GCD GCD.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace GCD
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             Console.Write("Enter the first number: ");
16             int n1 = Convert.ToInt32(Console.ReadLine());
17             Console.Write("Enter the second number: ");
18             int n2 = Convert.ToInt32(Console.ReadLine());
19
20             while (n1 != n2)
21             {
22                 if (n1 > n2)
23                     n1 = n1 - n2;
24                 else
25                     n2 = n2 - n1;
26             }
27             Console.WriteLine("G.C.D. of the above two numbers is " + n1);
28             Console.ReadLine();
29         }
30     }
```



## Output:

```
Enter the first number: 35  
Enter the second number: 21  
G.C.D. of the above two numbers is 7
```

#### 4. Program to check the prime number.

```
C# Prime Number Prime_Number.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Prime_Number
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             int a = 0;
16             Console.WriteLine("Enter a number:");
17             int n = Convert.ToInt32(Console.ReadLine());
18             for (int i = 1; i <= n; i++)
19             {
20                 if (n % i == 0)
21                 {
22                     a++;
23                 }
24             }
25             if (a == 2)
26             {
27                 Console.WriteLine("Prime Number");
28             }
29             else
30             {
31                 Console.WriteLine("Not a Prime Number");
32             }
33             Console.ReadLine();
34         }
35     }
36 }
```

### Output 1:

```
Enter a number:  
59  
Prime Number
```

### Output 2:

```
Enter a number:  
24  
Not a Prime Number
```

## 5. Program to print Fabonacci Series.

```
C# Fabonacci_Series Fabonacci_Series.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Fabonacci_Series
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             int val1 = 0, val2 = 1, val3, i, n;
14             Console.WriteLine("Enter the number of terms:");
15             n = Convert.ToInt32(Console.ReadLine());
16             Console.WriteLine("Fibonacci Series:");
17             Console.Write(val1 + " " + val2 + " ");
18             for (i = 2; i < n; ++i)
19             {
20                 val3 = val1 + val2;
21                 Console.Write(val3 + " ");
22                 val1 = val2;
23                 val2 = val3;
24             }
25             Console.ReadLine();
26         }
27     }
28 }
29
```

## Output:

```
Enter the number of terms:  
12  
Fibonacci Series:  
0 1 1 2 3 5 8 13 21 34 55 89
```

```
Enter the number of terms:  
8  
Fibonacci Series:  
0 1 1 2 3 5 8 13
```

```
Enter the number of terms:  
21  
Fibonacci Series:  
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
```

## 6. Program to print Half Pyramid Star Pattern.

```
Star Pattern 1 | Star_Pattern_1.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Star_Pattern_1
8  {
9      0 references
      class Program
10     {
11         0 references
        static void Main(string[] args)
12         {
13             for (int i = 1; i <= 15; i++)
14             {
15                 for (int j = 1; j <= i; j++)
16                 {
17                     Console.Write("*");
18                 }
19                 Console.WriteLine();
20             }
21             Console.ReadLine();
22         }
23     }
24 }
25
```

### Output:

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

## 7. Program to print Half Pyramid Number Pattern.

```
C# Number Pattern Number_Pattern.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Number_Pattern
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             int i, j, n, k = 1;
16             Console.Write("Enter number of rows: ");
17             n = Convert.ToInt32(Console.ReadLine());
18             for (i = 1; i <= n; i++)
19             {
20                 for (j = 1; j <= i; j++)
21                 {
22                     Console.Write("{0}", k++);
23                     Console.Write(" ");
24                 }
25                 Console.ReadLine();
26             }
27         }
28     }
29 }
```



### Output:

```
Enter number of rows: 6
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
```

## 8. Program to print Reverse Half Pyramid Star Pattern.

```
Star Pattern Star_Pattern.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Star_Pattern
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             for (int i = 16; i >= 1; i--)
16             {
17                 for (int j = 1; j <= i; j++)
18                 {
19                     Console.Write("*");
20                 }
21                 Console.WriteLine();
22             }
23             Console.ReadLine();
24         }
25     }
26 }
```

### Output:

[illegible]

## 9. Program to print Pyramid Star Pattern.

```
Star Pattern 2 | Star_Pattern_2.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Star_Pattern_2
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             for(int i = 1; i <= 20; i++)
16             {
17                 for (int j = 20; j >= i; j--)
18                 {
19                     Console.Write(" ");
20                 }
21                 for (int k = 1; k <= 2 * i - 1; k++)
22                 {
23                     Console.Write("*");
24                 }
25                 Console.WriteLine("");
26             }
27             Console.ReadLine();
28         }
29     }
```

### Output:

[illegible]

## 10. Program to print Reverse Pyramid Star Pattern.

```
C# Reverse Pyramid Pattern Reverse_Pyramid_Pattern.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Reverse_Pyramid_Pattern
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             for (int i = 1; i <= 15; i++)
16             {
17                 for (int j = 1; j < i; j++)
18                 {
19                     Console.Write(" ");
20                 }
21                 for (int j = 1; j <= (15 * 2 - (i * 2 - 1)); j++)
22                 {
23                     Console.Write("*");
24                 }
25                 Console.WriteLine();
26             }
27             Console.Read();
28         }
29     }
```

### Output:

[illegible]

## 11. Program to print Diamond Star Pattern.

```
C# Diamond Pattern Diamond_Pattern.Program
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Diamond_Pattern
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             Console.Write("Enter the number of rows: ");
16             int n = Convert.ToInt32(Console.ReadLine());
17
18             for (int i = 1; i <= n; i++)
19             {
20                 for (int j = 0; j < (n - i); j++)
21                     Console.Write(" ");
22                 for (int j = 1; j <= i; j++)
23                     Console.Write("*");
24                 for (int k = 1; k < i; k++)
25                     Console.Write("*");
26                 Console.WriteLine();
27             }
28
29             for (int i = n - 1; i >= 1; i--)
30             {
31                 for (int j = 0; j < (n - i); j++)
32                     Console.Write(" ");
33                 for (int j = 1; j <= i; j++)
34                     Console.Write("*");
35                 for (int k = 1; k < i; k++)
36                     Console.Write("*");
37                 Console.WriteLine();
38             }
39             Console.ReadLine();
40         }
41     }
42 }
```



### Output:

[illegible]

## 12. Program to print Pascal's Triangle.

```
Pascal Triangle Pascal_Triangle.Program
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Pascal_Triangle
8 {
9     0 references
10    class Program
11    {
12        0 references
13        static void Main(string[] args)
14        {
15            int rows = 10, val = 1, blank, i, j;
16            Console.WriteLine("Pascal's triangle");
17            for (i = 0; i < rows; i++)
18            {
19                for (blank = 1; blank <= rows - i; blank++)
20                    Console.Write(" ");
21                for (j = 0; j <= i; j++)
22                {
23                    if (j == 0 || i == 0)
24                        val = 1;
25                    else
26                        val = val * (i - j + 1) / j;
27                    Console.Write(val + " ");
28                }
29                Console.WriteLine();
30            }
31            Console.ReadLine();
32        }
33    }
34 }
```

## Output:

```
Pascal's triangle
  1
 1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
```

### 13. Program to compare two strings without using string library function.

```
C# Comp string Comp_string.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Comp_string
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             string str1, str2;
16             int flg = 0;
17             int i = 0, l1, l2, yn = 0;
18
19             Console.Write("Input the 1st string : ");
20             str1 = Console.ReadLine();
21
22             Console.Write("Input the 2nd string : ");
23             str2 = Console.ReadLine();
24
25             l1 = str1.Length;
26             l2 = str2.Length;
27
28             if (l1 == l2)
29             {
30                 for (i = 0; i < l1; i++)
31                 {
32                     if (str1[i] != str2[i])
33                     {
34                         yn = 1;
35                         i = l1;
36                     }
37                 }
38             }
39         }
40     }
41 }
```

```

37
38     if (l1 == l2)
39         flg = 0;
40     else if (l1 > l2)
41         flg = 1;
42     else if (l1 < l2)
43         flg = -1;
44
45     if (flg == 0)
46     {
47         if (yn == 0)
48             Console.Write("\nThe length of both strings are equal and \nalso, both strings are same.\n\n");
49         else
50             Console.Write("\nThe length of both strings are equal \nbut they are not same.\n\n");
51     }
52     else if (flg == -1)
53     {
54         Console.Write("\nThe length of the first string is smaller than second.\n\n");
55     }
56     else
57     {
58         Console.Write("\nThe length of the first string is greater than second.\n\n");
59     }
60     Console.ReadLine();
61 }
62 }
63

```

## Output:

```

Input the 1st string : aderfb
Input the 2nd string : hujio

The length of the first string is greater than second.

```

## 14. Program to count total number of alphabets, digits, and special characters in a string.

```
C# ADS count ADS_count.Program
7 namespace ADS_count
8 {
9     0 references
10    class Program
11    {
12        0 references
13        static void Main(string[] args)
14        {
15            string str;
16            int alp, digit, splch, i, l;
17            alp = digit = splch = i = 0;
18
19            Console.Write("Input the string : ");
20            str = Console.ReadLine();
21            l = str.Length;
22
23            while (i < l)
24            {
25                if ((str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z'))
26                {
27                    alp++;
28                }
29                else if (str[i] >= '0' && str[i] <= '9')
30                {
31                    digit++;
32                }
33                else
34                {
35                    splch++;
36                }
37                i++;
38            }
39
40            Console.WriteLine("\nNumber of Alphabets in the string is : {0}\n", alp);
41            Console.WriteLine("Number of Digits in the string is : {0}\n", digit);
42            Console.WriteLine("Number of Special characters in the string is : {0}\n\n", splch);
43            Console.ReadLine();
44        }
45    }
```

## Output:

```
Input the string : acvbg#$%&21348  
Number of Alphabets in the string is : 6  
Number of Digits in the string is : 5  
Number of Special characters in the string is : 4
```

## 15. Program to copy one string to another string.

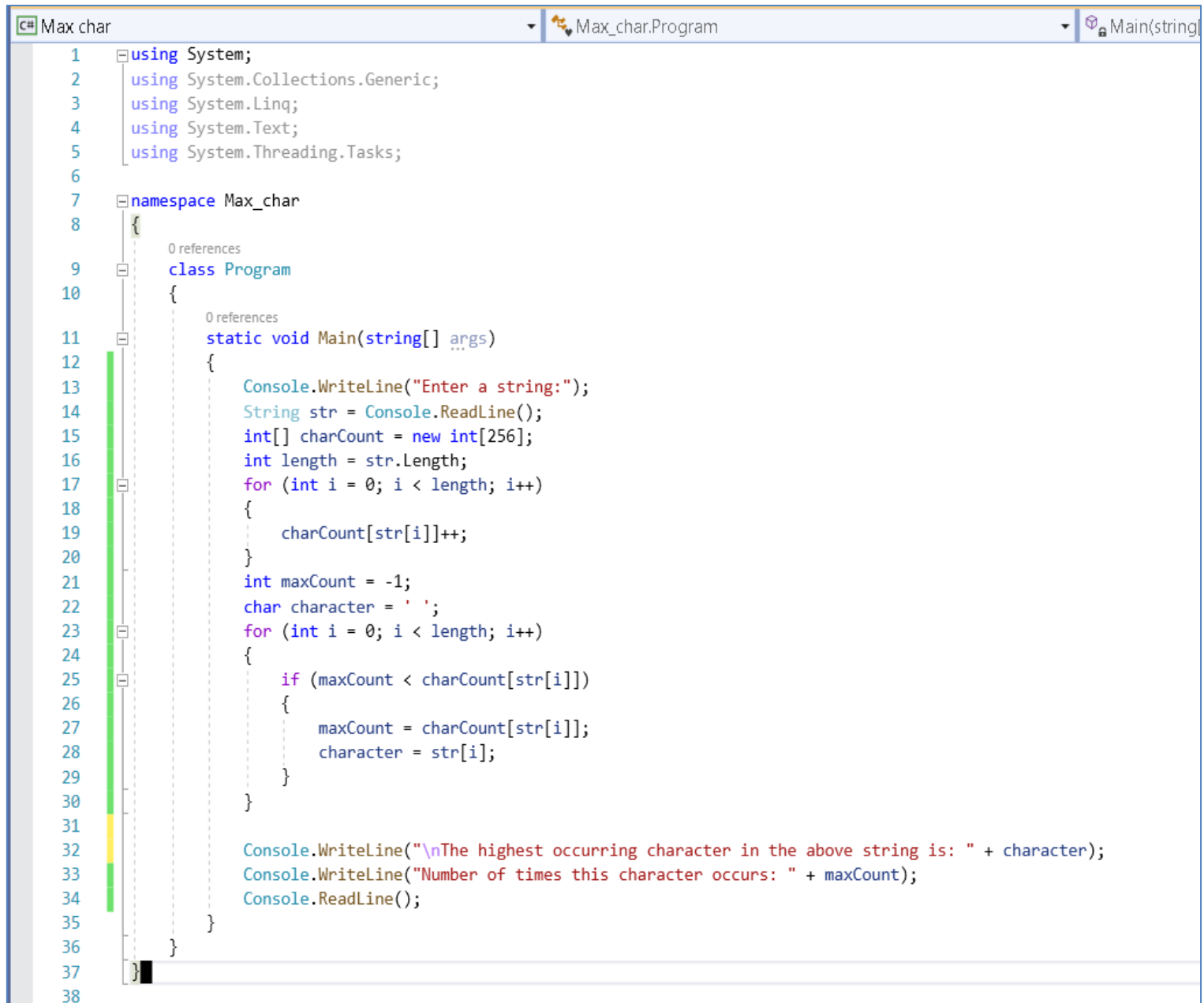
```
Copy String Copy_String.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Copy_String
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             string str1;
16             int i, l;
17
18             Console.Write("Input the string : ");
19             str1 = Console.ReadLine();
20
21             l = str1.Length;
22             string[] str2 = new string[l];
23
24             i = 0;
25             while (i < l)
26             {
27                 string tmp = str1[i].ToString();
28                 str2[i] = tmp;
29                 i++;
30             }
31             Console.WriteLine("The First string is : {0}\n", str1);
32             Console.WriteLine("The Second string is : {0}\n", string.Join("", str2));
33             Console.WriteLine("Number of characters copied : {0}\n\n", i);
34             Console.ReadLine();
35         }
36     }
37 }
```



## Output:

```
Input the string : dfrthy  
  
The First string is : dfrthy  
The Second string is : dfrthy  
Number of characters copied : 6
```

## 16. Program to find maximum occurring character in a string.



```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Max_char
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("Enter a string:");
14             String str = Console.ReadLine();
15             int[] charCount = new int[256];
16             int length = str.Length;
17             for (int i = 0; i < length; i++)
18             {
19                 charCount[str[i]]++;
20             }
21             int maxCount = -1;
22             char character = ' ';
23             for (int i = 0; i < length; i++)
24             {
25                 if (maxCount < charCount[str[i]])
26                 {
27                     maxCount = charCount[str[i]];
28                     character = str[i];
29                 }
30             }
31
32             Console.WriteLine("\nThe highest occurring character in the above string is: " + character);
33             Console.WriteLine("Number of times this character occurs: " + maxCount);
34             Console.ReadLine();
35         }
36     }
37 }
```

## Output:

```
Enter a string:  
dfreyubsfjf  
  
The highest occurring character in the above string is: f  
Number of times this character occurs: 3
```

## 17. Program to check whether a given substring is present in the given string.

```
C# Substring Substring.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Substring
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             string str1, str2;
16             bool m;
17
18             Console.Write("Input the string : ");
19             str1 = Console.ReadLine();
20
21             Console.Write("Input the substring to search : ");
22             str2 = Console.ReadLine();
23             m = str1.Contains(str2);
24
25             if (m)
26                 Console.WriteLine("The substring is in the string.\n\n");
27             else
28                 Console.WriteLine("The substring is not in the string. \n\n");
29                 Console.ReadLine();
30         }
31     }
32 }
```

### Output:

```
Input the string : abcdefgh
Input the substring to search : deg
The substring is not in the string.
```

```
Input the string : abcdefgh
Input the substring to search : def
The substring is in the string.
```

## 18. Program for Abstraction.

```
C# Abstraction Abstraction.Square
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Abstraction
8  {
9      2 references
10     abstract class Shape
11     {
12         2 references
13         public abstract int area();
14     }
15     2 references
16     class Square : Shape
17     {
18         private int side;
19         1 reference
20         public Square(int x = 0)
21         {
22             side = x;
23         }
24         2 references
25         public override int area()
26         {
27             Console.WriteLine("Area of Square: ");
28             return (side * side);
29         }
30     }
31     0 references
32     class Area
33     {
34         0 references
35         static void Main(string[] args)
36         {
37             Shape sh = new Square(4);
38             double result = sh.area();
39
40             Console.WriteLine("{0}", result);
41             Console.ReadLine();
42         }
43     }
44 }
```

**Output:**

```
Area of Square: 16
```

## 19. Program for Single Inheritance.

```
C# Single Inheritance Single_Inheritance.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Single_Inheritance
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             Father f = new Father();
14             f.Display();
15
16             Son s = new Son();
17             s.Display();
18             s.DisplayOne();
19
20             Console.ReadKey();
21         }
22     }
23     class Father
24     {
25         public void Display()
26         {
27             Console.WriteLine("Display");
28         }
29     }
30     class Son : Father
31     {
32         public void DisplayOne()
33         {
34             Console.WriteLine("DisplayOne");
35         }
36     }
```



**Output:**

```
Display  
Display  
DisplayOne
```

## 20. Program for Multilevel Inheritance.

```
C# Mult Inheritance Mult_Inheritance.Son
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Mult_Inheritance
8  {
9      2 references
10     class Son : Father
11     {
12         1 reference
13         public void DisplayTwo()
14         {
15             Console.WriteLine("Son.");
16         }
17         0 references
18         static void Main(string[] args)
19         {
20             Son s = new Son();
21             s.Display();
22             s.DisplayOne();
23             s.DisplayTwo();
24             Console.Read();
25         }
26     }
27     1 reference
28     class Grandfather
29     {
30         1 reference
31         public void Display()
32         {
33             Console.WriteLine("Grandfather.");
34         }
35     }
36     1 reference
37     class Father : Grandfather
38     {
39         1 reference
40         public void DisplayOne()
41         {
42             Console.WriteLine("Father.");
43         }
44     }
45 }
```

**Output:**

```
Grandfather.  
Father.  
Son.
```

## 21. Program for Multiple Inheritance.

```
C# Multiple Inheritance Multiple_Inheritance.RectangleDemo
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Multiple_Inheritance
8  {
9      1 reference
10     class Shape
11     {
12         1 reference
13         public void setWidth(int w)
14         {
15             width = w;
16         }
17         1 reference
18         public void setHeight(int h)
19         {
20             height = h;
21         }
22         protected int width;
23         protected int height;
24     }
25     1 reference
26     public interface PaintCost
27     {
28         2 references
29         int getCost(int area);
30     }
31     2 references
32     class Rectangle : Shape, PaintCost
33     {
34         2 references
35         public int getArea()
36         {
37             return (width * height);
38         }
39     }
```

```
C# Multiple Inheritance Multiple_Inheritance.RectangleDemo
32
33     2 references
34     public int getCost(int area)
35     {
36         return area * 80;
37     }
38     0 references
39     class RectangleDemo
40     {
41         0 references
42         static void Main(string[] args)
43         {
44             Rectangle Rect = new Rectangle();
45             int area;
46             Rect.setWidth(8);
47             Rect.setHeight(10);
48             area = Rect.getArea();
49             Console.WriteLine("Total area: {0}", Rect.getArea());
50             Console.WriteLine("Total paint cost: Rs. {0}", Rect.getCost(area));
51             Console.ReadKey();
52         }
53     }
54
```

**Output:**

```
Total area: 80
Total paint cost: Rs. 6400
```

## 22. Program for Method Overloading.

```
C# Overloading Overloading.Program
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Overloading
8  {
9      2 references
10     class Program
11     {
12         1 reference
13         public int Add(int num1, int num2)
14         {
15             return (num1 + num2);
16         }
17         1 reference
18         public int Add(int num1, int num2, int num3)
19         {
20             return (num1 + num2 + num3);
21         }
22         1 reference
23         public float Add(float num1, float num2)
24         {
25             return (num1 + num2);
26         }
27         1 reference
28         public string Add(string value1, string value2)
29         {
30             return (value1 + " " + value2);
31         }
32         0 references
33         static void Main(string[] args)
34         {
35             Program objProgram = new Program();
36             Console.WriteLine("Add with two int parameter :" + objProgram.Add(3, 2));
37             Console.WriteLine("Add with three int parameter :" + objProgram.Add(3, 2, 8));
38             Console.WriteLine("Add with two float parameter :" + objProgram.Add(3f, 22f));
39             Console.WriteLine("Add with two string parameter :" + objProgram.Add("hello", "world"));
40             Console.ReadLine();
41         }
42     }
43 }
```

## Output:

```
Add with two int parameter :5  
Add with three int parameter :13  
Add with two float parameter :25  
Add with two string parameter :hello world
```

## 23. Program for Method Overriding.

```
C# Overriding Overriding.Program
7 namespace Overriding
8 {
9     3 references
10    class BaseClass
11    {
12        3 references
13        public virtual int Add(int num1, int num2)
14        {
15            return (num1 + num2);
16        }
17    }
18    1 reference
19    class ChildClass : BaseClass
20    {
21        3 references
22        public override int Add(int num1, int num2)
23        {
24            if (num1 <= 0 || num2 <= 0)
25            {
26                Console.WriteLine("Enter First value : ");
27                num1 = Convert.ToInt32(Console.ReadLine());
28                Console.WriteLine("Enter Second value : ");
29                num2 = Convert.ToInt32(Console.ReadLine());
30            }
31            return (num1 + num2);
32        }
33    }
34    0 references
35    class Program
36    {
37        0 references
38        static void Main(string[] args)
39        {
40            BaseClass baseClassObj;
41            baseClassObj = new BaseClass();
42            Console.WriteLine("Base class Add :" + baseClassObj.Add(-3, 8));
43            baseClassObj = new ChildClass();
44            Console.WriteLine("Child class Add :" + baseClassObj.Add(-2, 2));
45            Console.ReadLine();
46        }
47    }
48 }
```



### Output:

```
Base class Add :5  
Enter First value :  
3  
Enter Second value :  
7  
Child class Add :10
```

## 24. Program for Interface.

```
C# Interface testClass
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  interface inter1
8  {
9      void display();
10 }
11
12 class testClass : inter1
13 {
14     public void display()
15     {
16         Console.WriteLine("Hey Everyone!!! This is me.");
17     }
18
19     public static void Main(String[] args)
20     {
21         testClass t = new testClass();
22         t.display();
23         Console.ReadLine();
24     }
25 }
26
```

## Output:

```
Hey Everyone!!! This is me.
```

## 25. Program for Exceptional Handling through try and catch.

```
C# Exp Handling Exp_Handling.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Exp_Handling
8  {
9      0 references
10     class Program
11     {
12         0 references
13         static void Main(string[] args)
14         {
15             try
16             {
17                 Console.Write("Enter number, a: ");
18                 int a = Convert.ToInt32(Console.ReadLine());
19
20                 Console.Write("Enter number, b: ");
21                 int b = Convert.ToInt32(Console.ReadLine());
22
23                 Console.WriteLine("a+b : " + (a + b));
24             }
25             catch (Exception ex)
26             {
27                 Console.WriteLine(ex.Message);
28             }
29             Console.WriteLine("Execution after try-catch block continues.");
30             Console.ReadLine();
31         }
32     }
```

## Output:

```
Enter number, a: 6  
Enter number, b: 8  
a+b : 14  
Execution after try-catch block continues.
```

## 26. Program for Properties.

```
C# Properties Properties.Program
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Properties
8  {
9      class User
10     {
11         private string name;
12         private string location;
13         public User(string a, string b)
14         {
15             name = a;
16             location = b;
17         }
18
19         public string Name
20         {
21             get
22             {
23                 return name;
24             }
25         }
26
27         public string Location
28         {
29             get
30             {
31                 return location;
32             }
33         }
34     }
35 }
```

C# Properties Properties.User

```
35
36 0 references
37 class Program
38 {
39     0 references
40     static void Main(string[] args)
41     {
42         User u = new User("Abcdef", "Ghijk");
43         Console.WriteLine("Name: " + u.Name);
44         Console.WriteLine("Location: " + u.Location);
45         Console.ReadLine();
46     }
47 }
```

Output:

```
Name: Abcdef
Location: Ghijk
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Properties
8  {
9      2 references
10     class User
11     {
12         1 reference
13         private string name;
14         public string Name
15         {
16             set
17             {
18                 name = value;
19             }
20         }
21
22         private string location;
23         1 reference
24         public string Location
25         {
26             set
27             {
28                 location = value;
29             }
30         }
31
32         1 reference
33         public void GetUserDetails()
34         {
35             Console.WriteLine("Name: " + name);
36             Console.WriteLine("Location: " + location);
37         }
38     }
39 }
```



C# Properties Properties.Program

35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49

0 references

class Program

{

0 references

static void Main(string[] args)

{

User u = new User();

u.Name = "Abcdef";

u.Location = "Ghijk";

u.GetUserDetails();

Console.ReadLine();

}

}

}

Output:

```
Name: Abcdef
Location: Ghijk
```

## 27. Program for Threading.

```
C# Threading Program
1  using System;
2  using System.Threading;
   0 references
3  class Program
4  {
   0 references
5      public static void Main()
6      {
7          Thread ThreadObject1 = new Thread(Example1);
8          Thread ThreadObject2 = new Thread(Example2);
9          ThreadObject1.Start();
10         ThreadObject2.Start();
11     }
   1 reference
12     static void Example1()
13     {
14         Console.WriteLine("Thread1 Started");
15         for (int i = 0; i <= 5; i++)
16         {
17             Console.WriteLine("Thread1 Executing");
18             Thread.Sleep(1000);
19         }
20     }
   1 reference
21     static void Example2()
22     {
23         Console.WriteLine("Thread2 Started");
24         for (int i = 0; i <= 5; i++)
25         {
26             Console.WriteLine("Thread2 Executing");
27             Thread.Sleep(1000);
28         }
29     }
30 }
31
```

## Output:

```
Thread2 Started  
Thread2 Executing  
Thread1 Started  
Thread1 Executing  
Thread2 Executing  
Thread1 Executing  
Thread2 Executing  
Thread1 Executing  
Thread1 Executing  
Thread2 Executing  
Thread1 Executing  
Thread2 Executing  
Thread1 Executing  
Thread2 Executing
```

## 28. Program for namespace.

```

C# Namespace TestClass
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace first_space
8  {
9      2 references
10     class namespace_cl
11     {
12         1 reference
13         public void func()
14         {
15             Console.WriteLine("Inside first_space");
16         }
17     }
18
19     namespace second_space
20     {
21         2 references
22         class namespace_cl
23         {
24             1 reference
25             public void func()
26             {
27                 Console.WriteLine("Inside second_space");
28             }
29         }
30     }
31
32     0 references
33     class TestClass
34     {
35         0 references
36         static void Main(string[] args)
37         {
38             first_space.namespace_cl fc = new first_space.namespace_cl();
39             second_space.namespace_cl sc = new second_space.namespace_cl();
40             fc.func();
41             sc.func();
42             Console.ReadKey();
43         }
44     }
45 }

```

## Output:

```
Inside first_space  
Inside second_space
```