TABLE OF CONTENTS

Table of Contents

1. Readkey and Rendering 2
2. 5 5 5 5 5 3

4 4 4 4 3

3 3 3 3

2 2 3

1 3

Generate this 3

3. loops in c# a pyramidical stars 4

* 4

*** 4

***** 4

******* 4

4. Diamond shape or double pyramid in csharp 6

* 6

*** 6

***** 6

******* 6

***** 6

*** 6

^ 6

5. Array iterator in for loop 7
6. Foreach loop iteration 8
7. Nestd loops with user input 9
8. While loop and user input 10
9. String Methods in c# 11
10. dice roller 12
11. random floating point genrator 13
12. biggest smallest average sum in cshartp 14
13. Jagged array in C# 16

14. sorting an array.....18
15. Transposition of matrix.....20

1. Readkey and Rendering

```
using System;

namespace HelloWorld
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("*****");
            Console.WriteLine("****");
            Console.WriteLine("***");
            Console.WriteLine("**");
            Console.WriteLine("*");
            Console.ReadKey();
        }
    }
}
```

2. 5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
Generate this

```
using System;

class Program
{
    static void Main()
    {
        for (int i = 5; i >= 1; i--)
        {
            for (int j = 1; j <= i; j++)
            {
                Console.Write(i +"\t");
            }
            Console.WriteLine();
            Console.WriteLine();

        }
    }
}
```

3. loops in c# a pyramidical stars

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

using System;

```
class Program
{
    static void Main()
    {

            int n = 28;

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

            {

                for (int j = 0; j < 2 * n; j++)

                {
                    if (j >= n  - i && j <= n  + i)
                    {
```

```
                    Console.Write("*");

                }

                else
                {

                    Console.Write(" ");

                }

            }

            Console.WriteLine("\n");

        }
    }
}
```

4. Diamond shape or double pyramid in csharp

```
   *
  ***
 *****
*******
 *****
  ***
   ^
```

```
using System;

class Program
{
    static void Main()
    {
        int n = 8;

        // Print the upper pyramid
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < 2 * n; j++)
            {
                if (j >= n - i && j <= n + i)
                    Console.Write("*");
                else
```

```
                Console.Write(" ");
            }
            Console.WriteLine();
        }

        // Print the lower inverse pyramid
        for (int i = n - 2; i >= 0; i--)
        {
            for (int j = 0; j < 2 * n; j++)
            {
                if (j >= n - i && j <= n + i)
                    Console.Write("*");
                else
                    Console.Write(" ");
            }
            Console.WriteLine();
        }
    }
}
```

5. Array iterator in for loop

```
using System;

class Program
```

```
{
    static void Main()
    {
        String[] cars =
["Mercedes","Lexus","Toyota","Honda"];
        for (int i = 0; i < cars.Length; i++) {
            Console.Write(i+1+": ");
            Console.WriteLine(cars[i]);



        }
    }
}
```

6. *Foreach loop iteration*

```
using System;

class Program
{
    static void Main()
    {
```

```
        String[] cars =
["Mercedes","Lexus","Toyota","Honda"];
        foreach (String car in cars) {
            Console.WriteLine(car);
        }
    }
}
```

7. Nestd loops with user input

```
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("how many rows? ");
        int rows = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("how many columns? ");
        int cols = Convert.ToInt32(Console.ReadLine());
```

```
        Console.WriteLine("what symbol? ");
        int sym = Convert.ToInt32(Console.ReadLine());


        for(int i = 0; i < rows; i++)
        {
            for(int j = 0; j < cols; j++)
            {
                Console.Write(sym+" ");
            }
            Console.WriteLine();
        }

    }
}
```

8. While loop and user input

```
using System;

class Program
{
    static void Main()
    {
```

```
        string name = "";
        while (name == "")
        {
            Console.WriteLine("enter your name: ");
            name= Console.ReadLine();

        }
        Console.WriteLine("Hello " + name);
        Console.ReadKey();

    }
}
```

9. String Methods in c#

```
using System;

class Program
{
    static void Main()
    {
        String name = "Rustam Shrestha";
        name = name.ToLower();
        name = name.ToUpper();
        Console.WriteLine(name);
```

```
        //this ffollowing not working
        String phone = "01-13841-66";
        String newphone =phone.Replace("-", "");
        Console.WriteLine(phone);

        String fname = "Rustam";
        String fnameMod = fname.Insert(0, "Mr. ");
        Console.WriteLine(fnameMod);
        String extractedName = fnameMod.Substring(0, 3);
        Console.WriteLine(extractedName);
        Console.ReadKey();


    }
}
```

10. dice roller

```
using System;

class Program
{
    static void Main()
```

```
    {
        Random rnd = new Random();
        for (int i = 0; i < 10; i++) {
        int num = rnd.Next(1, 7);
        Console.WriteLine(num);
        }
        Console.ReadKey();


    }
}
```

11. *random floating point genrator*

```
using System;

class Program
{
    static void Main()
    {
        Random rnd = new Random();
        for (int i = 0; i < 10; i++) {
        double num = rnd.NextDouble();
        Console.WriteLine(num);
```

```
        }
        Console.ReadKey();


    }
}
```

12. *biggest smallest average sum in cshartp*

```
using System;

class Program
{
    static void Main()
    {
        int n, i, biggest, smallest, sum = 0;
        int[] a = new int[100];
        Console.WriteLine("Enter the size of the array:");
        n = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter elements in the array:");
        for (i = 0; i < n; i++)
        {
            a[i] = Convert.ToInt32(Console.ReadLine());
        }
```

```
        biggest = a[0];
        smallest = a[0];
        for (i = 0; i < n; i++)
        {
            if (a[i] > biggest)
            {
                biggest = a[i];
            }
            if (a[i] < smallest)
            {

                smallest = a[i];
            }
            sum = sum + a[i];
        }
            Console.WriteLine("highest element is "+
biggest);
            Console.WriteLine("smallest element is "+
smallest);
            Console.WriteLine("sum element is "+ sum);

        }
    }
```

13. *Jagged array in C#*

```
using System;

namespace JaggedArrayExample
{
    class Program
    {
        static void Main(string[] args)
        {
            // Get the number of rows from user input
            Console.Write("Enter the number of rows: ");
            int numRows = int.Parse(Console.ReadLine());

            // Create a jagged array with the specified
number of rows
            int[][] jaggedArray = new int[numRows][];

            // Initialize each row of the jagged array
            for (int i = 0; i < numRows; i++)
            {
                Console.Write($"Enter the number of elements
for row {i + 1}: ");
                int numElements =
int.Parse(Console.ReadLine());
```

```
                // Create an array for the current row
                jaggedArray[i] = new int[numElements];

                // Initialize each element of the current
row
                for (int j = 0; j < numElements; j++)
                {
                    Console.Write($"Enter element {j + 1}
for row {i + 1}: ");
                    jaggedArray[i][j] =
int.Parse(Console.ReadLine());
                }
            }

            // Display the elements of the jagged array
            Console.WriteLine("\nJagged Array Elements:");
            for (int i = 0; i < numRows; i++)
            {
                for (int j = 0; j < jaggedArray[i].Length;
j++)
                {
                    Console.Write(jaggedArray[i][j] + " ");
                }
                Console.WriteLine();
            }
```

```
            Console.ReadLine();
        }
    }
}
```

14. *sorting an array*

```
using System;

namespace BubbleSortExample
{
    class Program
    {
        static void Main(string[] args)
        {
            // Step 1: Accept user input for the number of
elements
            Console.Write("Enter the number of elements: ");
            int n = int.Parse(Console.ReadLine());
```

```
            // Step 2: Take user input for the elements and
store them in an array
            int[] arr = new int[n];
            for (int i = 0; i < n; i++)
            {
                Console.Write($"Enter element {i + 1}: ");
                arr[i] = int.Parse(Console.ReadLine());
            }

            // Step 3: Sort the array using bubble sort
            for (int i = 0; i < n - 1; i++)
            {
                for (int j = 0; j < n - i - 1; j++)
                {
                    if (arr[j] > arr[j + 1])
                    {
                        // Swap elements
                        int temp = arr[j];
                        arr[j] = arr[j + 1];
                        arr[j + 1] = temp;
                    }
                }
            }

            // Display the sorted array
```

```
            Console.WriteLine("\nSorted Array (Ascending
Order):");
            foreach (int num in arr)
            {
                Console.Write(num + " ");
            }

            Console.ReadLine();
        }
    }
}
```

15. *Transposition of matrix*

```
using System;

class Program
{
    static void Main()
    {
        // Get user input for matrix dimensions (rows and
columns)
        Console.Write("Enter the number of rows: ");
        int rows = int.Parse(Console.ReadLine());
```

```
        Console.Write("Enter the number of columns: ");
        int cols = int.Parse(Console.ReadLine());

        // Initialize an empty matrix
        int[,] matrix = new int[rows, cols];

        // Get user input for matrix elements
        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                Console.Write($"Enter element at position
({i + 1}, {j + 1}): ");
                matrix[i, j] =
int.Parse(Console.ReadLine());
            }
        }

        // Initialize an empty transposed matrix
        int[,] transposedMatrix = new int[cols, rows];

        // Perform transpose operation
        for (int j = 0; j < cols; j++)
        {
            for (int i = 0; i < rows; i++)
```

```
            {
                transposedMatrix[j, i] = matrix[i, j];
            }
        }

        // Display the transposed matrix
        Console.WriteLine("Transposed Matrix:");
        for (int i = 0; i < cols; i++)
        {
            for (int j = 0; j < rows; j++)
            {
                Console.Write(transposedMatrix[i, j] + " ");
            }
            Console.WriteLine();
        }
    }
}
```