

DAY13 ASSIGNMENT

(9th FEB 2022)

BY RAM CHARAN

1. Declare a 2 dimensional array of size (2, 2) and initialize using indexes and print values using nested for loop.

Code:

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

namespace Day13Project1
{
    //Author :Rc
    //Purpose: Declare 2 dimensional array
    internal class Program
    {
        static void Main(string[] args)
        {

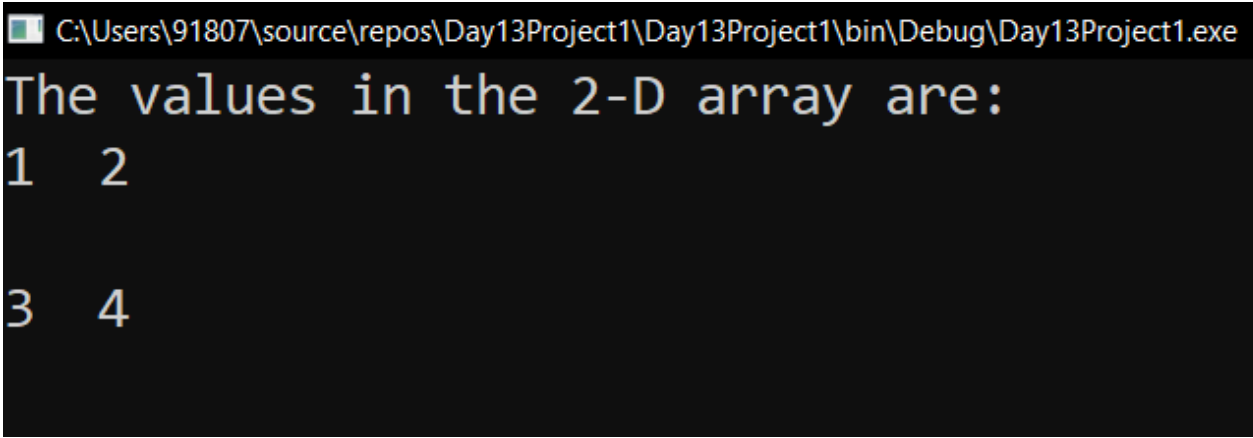
            //2 Dimensional array creation
            int[,] data = new int[2,2];
            //Initialising using indexes
            data[0,0] = 1;
            data[0,1] = 2;
            data[1,0] = 3;
            data[1,1] = 4;
            //Printing values of 2-D array
            for(int i=0;i<2;i++)
```

```

    {
        for(int j=0;j<2;j++)
        {
            Console.Write(data[i,j]+" ");
        }
        Console.WriteLine("\n");
    }
    Console.ReadLine();
}
}
}

```

Output:



```

C:\Users\91807\source\repos\Day13Project1\Day13Project1\bin\Debug\Day13Project1.exe
The values in the 2-D array are:
1  2
3  4

```

2. Declare a 2-D array of size (3,2) and initialize in same line while declaring and print the values using nested for loop.

Code:

```

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

namespace Day13Project2
{
    //Author : Rc
    /*Purpose: Declare 2-D of (3,2)*/
    internal class Program
    {
        static void Main(string[] args)
        {

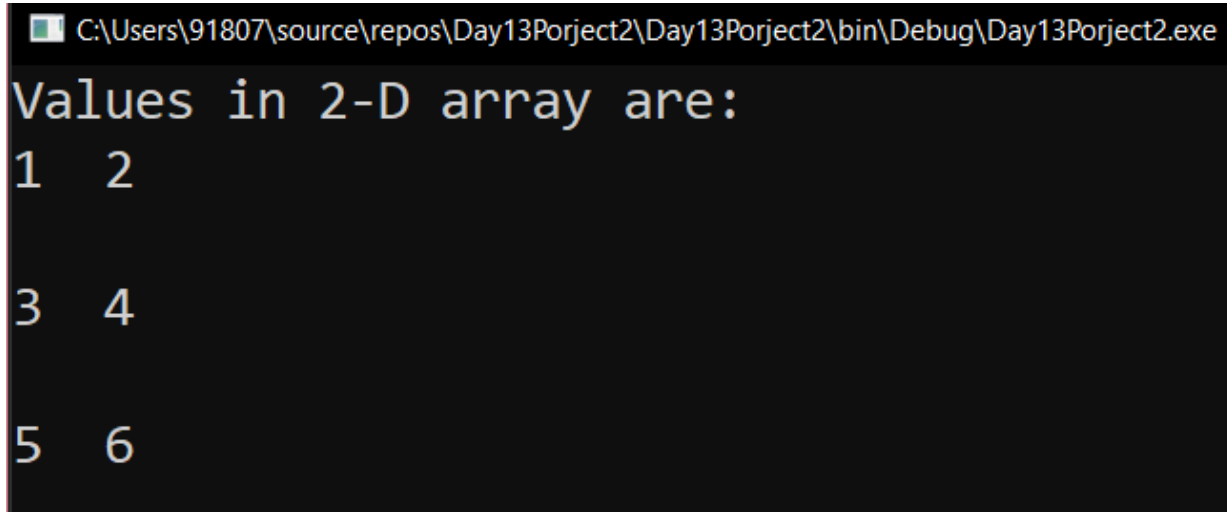
```

```

//2-D Array creation
int[,] data = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 } };
//Printing values in 2-D array
Console.WriteLine("Values in 2-D array are:");
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 2; j++)
    {
        Console.Write(data[i, j] + " ");
    }
    Console.WriteLine("\n");
}
Console.ReadLine();
}
}

```

Output:



```

C:\Users\91807\source\repos\Day13Porject2\Day13Porject2\bin\Debug\Day13Porject2.exe
Values in 2-D array are:
1 2
3 4
5 6

```

3. Declare a 2-D array of size (3,3) and print trace of the array.

Code:

```

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

namespace Day13Project3
{

```

```

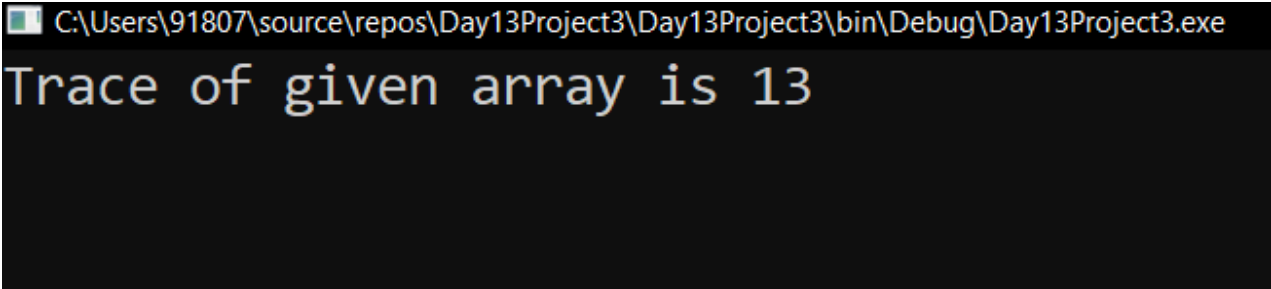
//Author : Rc
/*Purpose: Declare a 2-D array of size(3,3) and print trace of array*/
internal class Program
{
    static void Main(string[] args)
    {
        //2-D array creation and initialisation
        int[,] data = new int[,] { { 1,2,3 }, { 4,5,6 }, { 5,6,7 } };
        int sum = 0;

        //printing trace of 2-D array

        for (int i = 0; i < 3; i++) //rows
        {
            for (int j = 0; j < 3; j++) //columns
            {
                if(i==j) //Condition for finding trace
                {
                    sum = sum + data[i, j];
                }
            }
        }
        Console.WriteLine("Trace of given array is {0}",sum);
        Console.ReadLine();
    }
}

```

Output:



```

C:\Users\91807\source\repos\Day13Project3\Day13Project3\bin\Debug\Day13Project3.exe
Trace of given array is 13

```

4. Declare a 2-D array of size (2,2) and read values from user and print array values.

Code:

```

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

namespace Day13Project4
{
    //Author: Rc
    /*Purpose: Declare a 2-D array and read values from user*/
    internal class Program
    {
        static void Main(string[] args)
        {
            //2-D array creation
            int[,] data = new int[2, 2];
            //Reading user input
            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.WriteLine($"Enter array value at ({i},{j}) : ");
                    data[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
            Console.WriteLine("2-D array is : ");
            //printing user input
            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.Write(data[i,j]+" ");
                }
                Console.WriteLine("\n");
            }

            Console.ReadLine();
        }
    }
}

```

Output:

C:\Users\91807\source\repos\Day13Project4\Day13Project4\bin\Debug\Day13Project4.exe

Enter array value at (0,0) :

1

Enter array value at (0,1) :

3

Enter array value at (1,0) :

6

Enter array value at (1,1) :

1

2-D array is :

1 3

6 1

5. Declare two 2-D arrays of size(2,2) and read values from user and print sum of the two matrices.

Code:

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

namespace Day13Project5
{
    //Author : Rc
    /*Purpose : Declare two 2-D arrays and find sum*/
    internal class Program
    {
        static void Main(string[] args)
        {
            //2-D Array1
            int[,] a = new int[2, 2];
            //2-D Array 2
            int[,] b = new int[2, 2];
```

```

int[,] sum = new int[2,2];
//Reading user input for matrix a
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        Console.WriteLine($"Enter array value at ({i},{j}) : ");
        a[i, j] = Convert.ToInt32(Console.ReadLine());
    }
}
Console.WriteLine("Matrix a is : ");
//printing user input of matrix a
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        Console.Write(a[i, j] + " ");
    }
    Console.WriteLine("\n");
}
//Reading user input for matrix b
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        Console.WriteLine($"Enter array value at ({i},{j}) : ");
        b[i, j] = Convert.ToInt32(Console.ReadLine());
    }
}
Console.WriteLine("Matrix b is : ");
//printing user input of matrix b
for (int i = 0; i < 2; i++)
{
    for (int j = 0; j < 2; j++)
    {
        Console.Write(b[i, j] + " ");
    }
    Console.WriteLine("\n");
}
Console.WriteLine("After addition Matrix is :");
for(int i = 0;i < 2; i++)
{
    for(int j = 0;j < 2; j++)
    {
        sum[i,j]=a[i,j]+b[i,j];

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

```

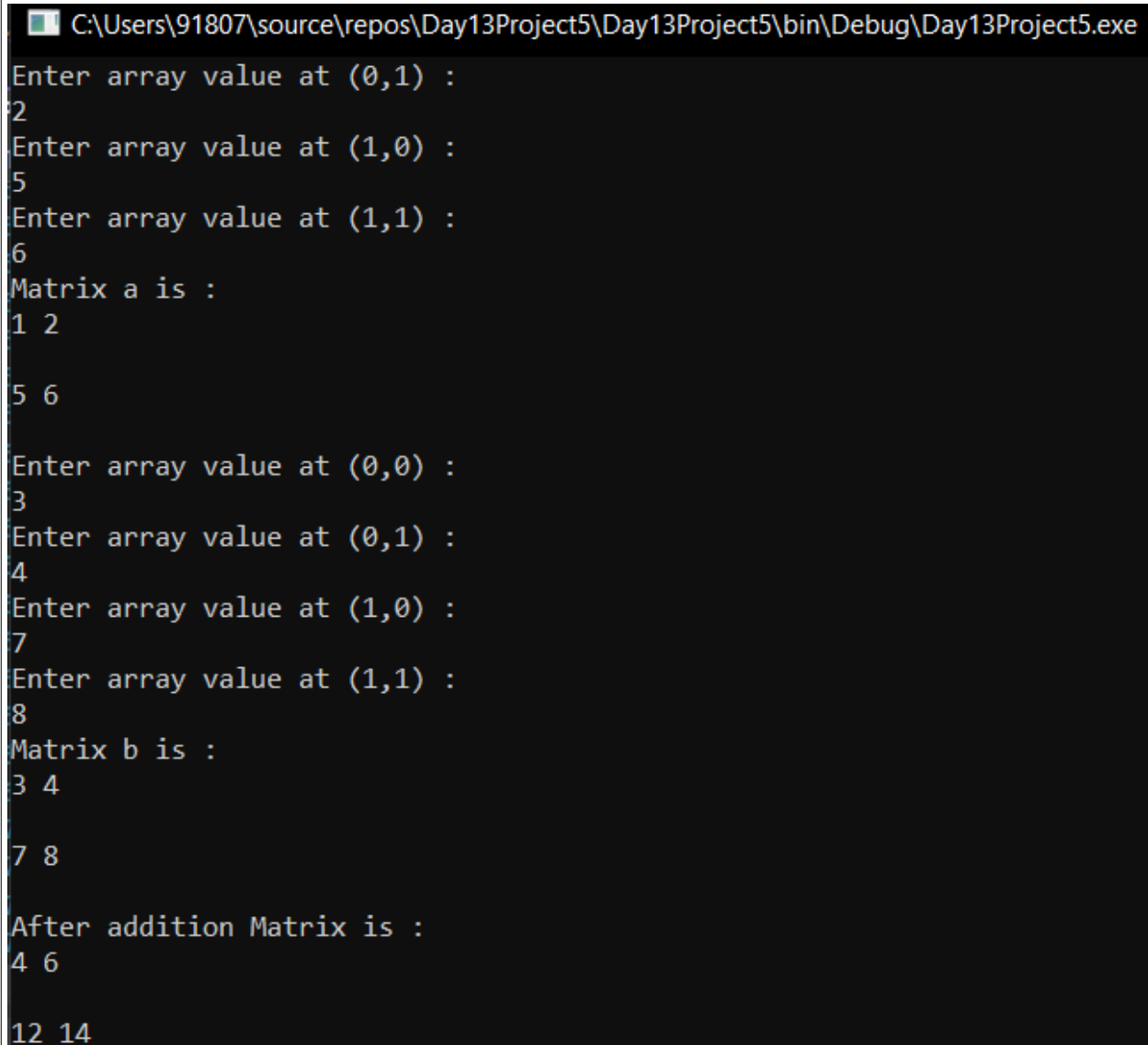
```

        Console.WriteLine("\n");
    }

    Console.ReadLine();
}
}
}

```

Output:



```

C:\Users\91807\source\repos\Day13Project5\Day13Project5\bin\Debug\Day13Project5.exe
Enter array value at (0,1) :
2
Enter array value at (1,0) :
5
Enter array value at (1,1) :
6
Matrix a is :
1 2
5 6
Enter array value at (0,0) :
3
Enter array value at (0,1) :
4
Enter array value at (1,0) :
7
Enter array value at (1,1) :
8
Matrix b is :
3 4
7 8
After addition Matrix is :
4 6
12 14

```

6. Declare two 2-D arrays of size(2,2) and read values from user and print the product of the two matrices.

Code:

```

using System;
using System.Collections.Generic;
using System.Linq;

```



```

using System.Text;
using System.Threading.Tasks;

namespace Day13Project6
{
    //Author: Rc
    /*****Purpose:Matrix Multiplication*****/
    internal class Program
    {
        static void Main(string[] args)
        {
            //Reading rows and columns for Matrix A

            Console.WriteLine("Enter Rows for Matrix a");
            int m=Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Columns for Matrix a");
            int n = Convert.ToInt32(Console.ReadLine());
            //Array of Matrix A

            int[,] a = new int[m,n];

            //Reading rows and columns for Matrix B
            Console.WriteLine("Enter Rows for Matrix b");
            int p = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Columns for Matrix b");
            int q = Convert.ToInt32(Console.ReadLine());
            //Array of Matrix B
            int[,] b = new int[p,q];

            //int[,] c = new int[2,2];

            //Reading user input for Matrix A
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    Console.WriteLine($"Enter array value at ({i},{j}) : ");
                    a[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
            Console.WriteLine("Matrix a is : ");
            //printing user input for Matrix A
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    Console.Write(a[i, j] + " ");
                }
            }
        }
    }
}

```

```

        Console.WriteLine("\n");
    }
    //Reading user input for B
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < q; j++)
        {
            Console.WriteLine($"Enter array value at ({i},{j}) : ");
            b[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }
    Console.WriteLine("Matrix b is : ");
    //printing user input for B
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < q; j++)
        {
            Console.Write(b[i, j] + " ");
        }
        Console.WriteLine("\n");
    }
    //Multiplication

    if(n==p)
    {
        int[,] c = new int[m, q]; //final Matrix
        Console.WriteLine("After Matrix Multiplication:");
        for(int i = 0; i < m; i++)
        {
            for (int j = 0; j < q; j++)
            {
                c[i, j] = 0;

                for (int k = 0; k < n; k++)
                {
                    c[i, j] += a[i, k] * b[k, j];
                }
                Console.Write(c[i, j] + " "); //Printing Matrix
            }
            Console.WriteLine();
        }
    }
    else
    {
        Console.WriteLine("Matrix multiplication not possible");
    }
}

```

```

        Console.ReadLine();
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day13Project6
{
    //Author: Rc
    /*****Purpose:Matrix Multiplication*****/
    internal class Program
    {
        static void Main(string[] args)
        {
            //Reading rows and columns for Matrix A

            Console.WriteLine("Enter Rows for Matrix a");
            int m=Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Columns for Matrix a");
            int n = Convert.ToInt32(Console.ReadLine());
            //Array of Matrix A

            int[,] a = new int[m,n];

            //Reading rows and columns for Matrix B
            Console.WriteLine("Enter Rows for Matrix b");
            int p = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Enter Columns for Matrix b");
            int q = Convert.ToInt32(Console.ReadLine());
            //Array of Matrix B
            int[,] b = new int[p,q];

            //int[,] c = new int[2,2];

            //Reading user input for Matrix A
            for (int i = 0; i < m; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    Console.WriteLine($"Enter array value at ({i},{j}) : ");
                    a[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
            Console.WriteLine("Matrix a is : ");

```

```

//printing user input for Matrix A
for (int i = 0; i < m; i++)
{
    for (int j = 0; j < n; j++)
    {
        Console.Write(a[i, j] + " ");
    }
    Console.WriteLine("\n");
}
//Reading user input for B
for (int i = 0; i < p; i++)
{
    for (int j = 0; j < q; j++)
    {
        Console.WriteLine($"Enter array value at ({i},{j}) : ");
        b[i, j] = Convert.ToInt32(Console.ReadLine());
    }
}
Console.WriteLine("Matrix b is : ");
//printing user input for B
for (int i = 0; i < p; i++)
{
    for (int j = 0; j < q; j++)
    {
        Console.Write(b[i, j] + " ");
    }
    Console.WriteLine("\n");
}
//Multiplication

if(n==p)
{
    int[,] c = new int[m, q]; //final Matrix
    Console.WriteLine("After Matrix Multiplication:");
    for(int i = 0; i < m; i++)
    {
        for (int j = 0; j < q; j++)
        {
            c[i, j] = 0;

            for (int k = 0; k < n; k++)
            {
                c[i, j] += a[i, k] * b[k, j];
            }
            Console.Write(c[i, j] + " "); //Printing Matrix
        }
        Console.WriteLine();
    }
}

```

```
    }  
  
    }  
    else  
    {  
        Console.WriteLine("Matrix multiplication not possible");  
    }  
    Console.ReadLine();  
}  
}
```

Output:

C:\Users\91807\source\repos\Day13Project6\Day13Project6\bin\Debug\Day13Project6.exe

Enter array value at (0,0) :

1

Enter array value at (0,1) :

2

Enter array value at (1,0) :

3

Enter array value at (1,1) :

4

Matrix a is :

1 2

3 4

Enter array value at (0,0) :

1

Enter array value at (0,1) :

2

Enter array value at (1,0) :

3

Enter array value at (1,1) :

4

Matrix b is :

1 2

3 4

After Matrix Multiplication:

7 10

15 22

7.What is Jagged array?

- Jagged array is an array whose elements of array are possibly of different sizes.

- Syntax:
 datatype[][] variablename= new datatype[rows][columns];
- Example:
 int[][] a = new int[5][];

What are Benefits of jagged array?

- It makes things easier where there is a need to store data in multi-dimensional array with same variable name.
- Memory wastage is reduced

8.WACP to declare a jagged array and print values.

Code:

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

namespace Day13Project7
{
    //Author : Rc
    //Purpose :Jagged arrays
    internal class Program
    {
        static void Main(string[] args)
        {
            //Jagged array creation
            int[][] d = new int[5][];

            //Initialising jagged arrays

            d[0]=new int[] {1,2,5};
            d[1]=new int[] {1};
            d[2]=new int[] {3,5};
            d[3]=new int[] {4,6};
            d[4]=new int[] {5,7,8,8,7,5};

            //Printing Jagged array
            Console.WriteLine("Jagged Array is:");

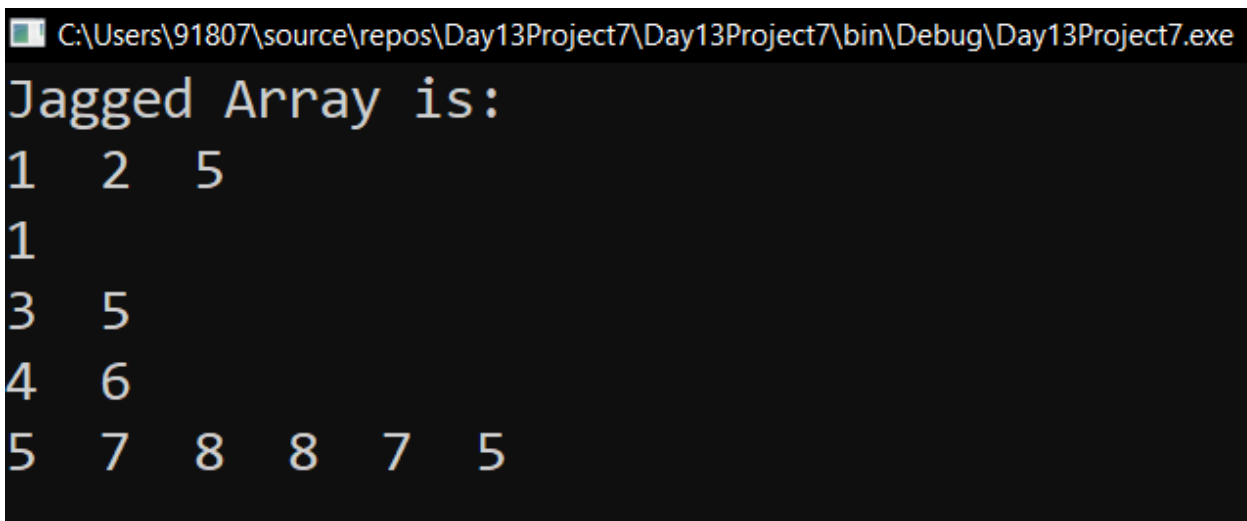
            for (int i = 0; i < d.Length; i++)
```

```

    {
        for(int j = 0;j<d[i].Length;j++)
        {
            Console.Write(d[i][j]+" ");
        }
        Console.WriteLine();
    }
    Console.ReadLine();
}
}
}

```

Output:



```

C:\Users\91807\source\repos\Day13Project7\Day13Project7\bin\Debug\Day13Project7.exe
Jagged Array is:
1  2  5
1
3  5
4  6
5  7  8  8  7  5

```

9.What is recursion?

➤ Recursion is nothing but process of calling function itself.

10.WACP to illustrate the use of Recursion.

Code:

```

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

namespace Day13Project8
{

```



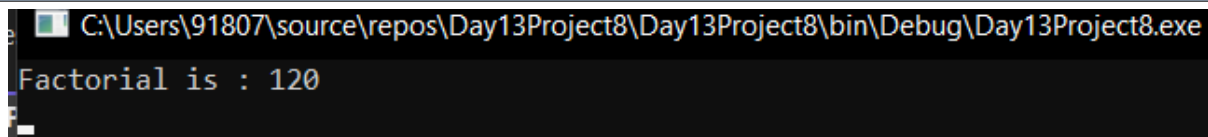
```

//Author:Rc
/*Purpose: Simple program for implementing recursion*/
public class Rc
{
    /// <summary>
    /// This method is to find factorial of given integer
    /// </summary>
    /// <param name="n"></param>
    /// <returns>Factorial</returns>
    public static int Fact(int n)
    {
        if (n == 0)
            return 1;
        else
            return n * Fact(n - 1); //Recursion
    }
}
internal class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Factorial is : {0} ",Rc.Fact(5));

        Console.ReadLine();
    }
}

```

Output:



What are benefits of Recursion?

- ◆ Recursion reduces time complexity.
- ◆ Reduces time to debug.
- ◆ Performs better in solving problems on tree structures.

11. Write couple of points on Stack.

- Stack represents a Last-in ,First-Out collection of object.
- It is used when we need last-in first-out access to items.
- When you add an element then it is known as Pushing.

- When you delete an element then it is known as Popping.
- When you need the top most element we can call it by using Peek()

WACP to illustrate Stack.

Code:

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

namespace Day13Project9
{
    //Author : Rc
    /**Purpose: Stack implementation**/
    internal class Program
    {
        static void Main(string[] args)
        {
            //Stack creation (Generics)
            Stack<int> s = new Stack<int>();
            //Adding elements into stack
            s.Push(10);
            s.Push(20);
            s.Push(30);
            Console.WriteLine("After push \n");
            Console.WriteLine("Total Count is : ");
            Console.WriteLine(s.Count()); //To get number of elements in stack
            Console.WriteLine("The Deleted element is:");
            Console.WriteLine(s.Pop());
            Console.WriteLine("Top element in stack is :");
            Console.WriteLine(s.Peek());
            Console.WriteLine("After Pop Count is :");
            Console.WriteLine(s.Count());

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

Output:

C:\Users\91807\source\repos\Day13Project9\Day13Project9\bin\Debug\Day13Project9.exe

After push

Total Count is :

3

The Deleted element is:

30

Top element in stack is :

20

After Pop Count is :

2

12. Write couple of points about Queue.

- Queue represents First-In-First-Out collection of objects.
- It is used when we need first-in first-out access of items.
- When you add an element into queue it is known as Enqueue
- When you delete an element from queue it is known as Dequeue.

WACP to illustrate Queue.

Code:

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

namespace Day13Project10
{
    //Author :Rc
    /*Purpose: Implementation of Queue*/
    internal class Program
    {
        static void Main(string[] args)
        {
            //Queue Declaration
        }
    }
}
```

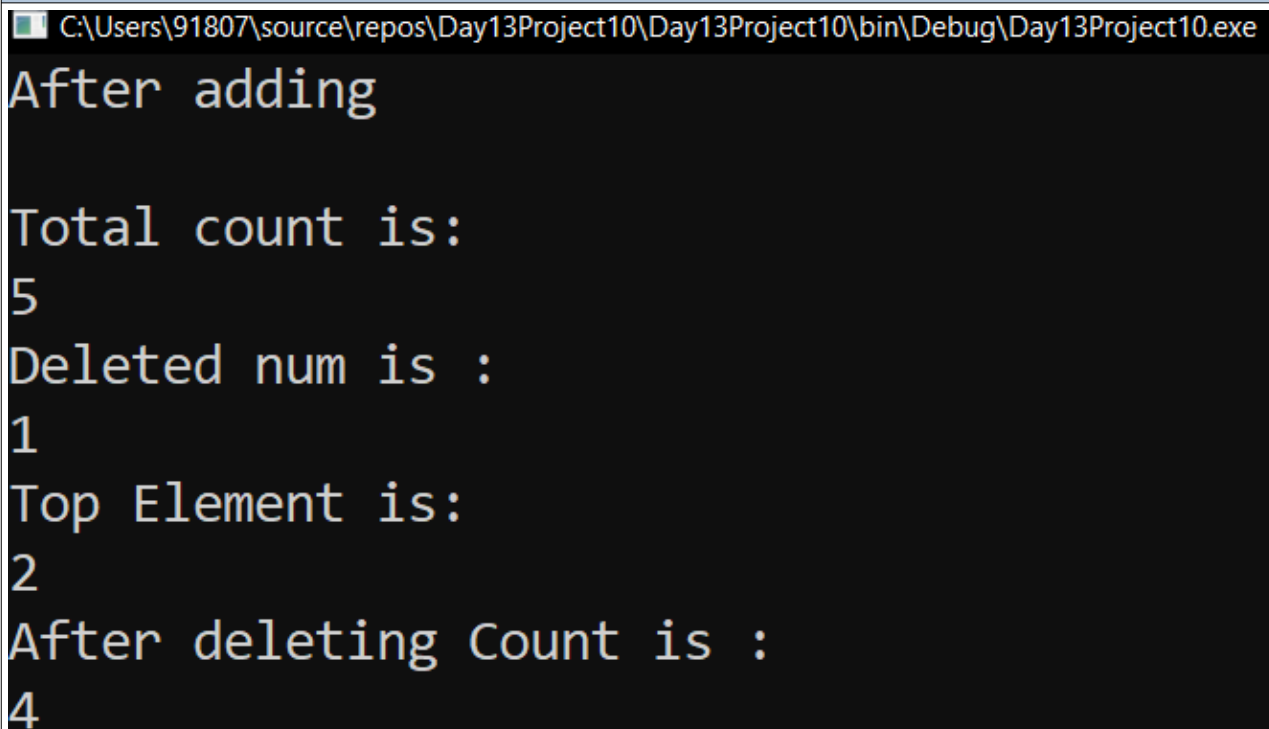
```

Queue<int> queue = new Queue<int>();
//Adding elements into queue
queue.Enqueue(1);
queue.Enqueue(2);
queue.Enqueue(3);
queue.Enqueue(4);
queue.Enqueue(5);
Console.WriteLine("After adding \n");
Console.WriteLine("Total count is:");
Console.WriteLine(queue.Count());
Console.WriteLine("Deleted num is :");
Console.WriteLine(queue.Dequeue()); // Deleting element from queue
Console.WriteLine("Top Element is:");
Console.WriteLine(queue.Peek());
Console.WriteLine("After deleting Count is :");
Console.WriteLine(queue.Count());

    Console.ReadLine();
}
}
}

```

Output:



```

C:\Users\91807\source\repos\Day13Project10\Day13Project10\bin\Debug\Day13Project10.exe
After adding

Total count is:
5
Deleted num is :
1
Top Element is:
2
After deleting Count is :
4

```