

Day 13  
Assignment  
By M Mary Margarette  
On 09-02-2022

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

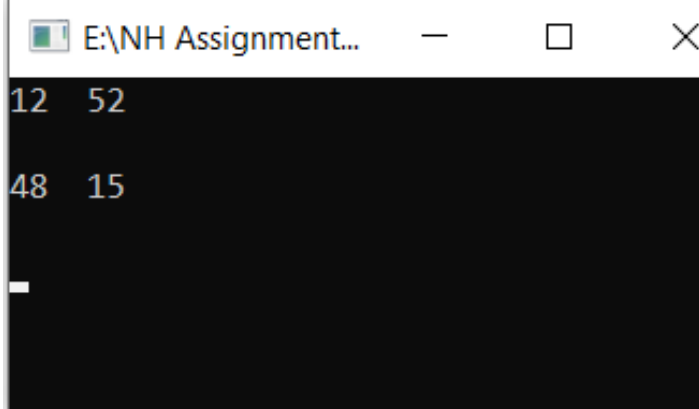
Code:

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

namespace Day_13_Project_1
{
    //Author: Mary Margaret
    //Two-Dimensional Array
    internal class Program
    {
        static void Main(string[] args)
        {
            //2-D Array Initialization
            int[,] data = new int[2, 2];
            data[0, 0] = 12;
            data[0, 1] = 52;
            data[1, 0] = 48;
            data[1, 1] = 15;

            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:



```
12 52
48 15
```

Declare a 2-D array of size (3,2) and initialize in the 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;

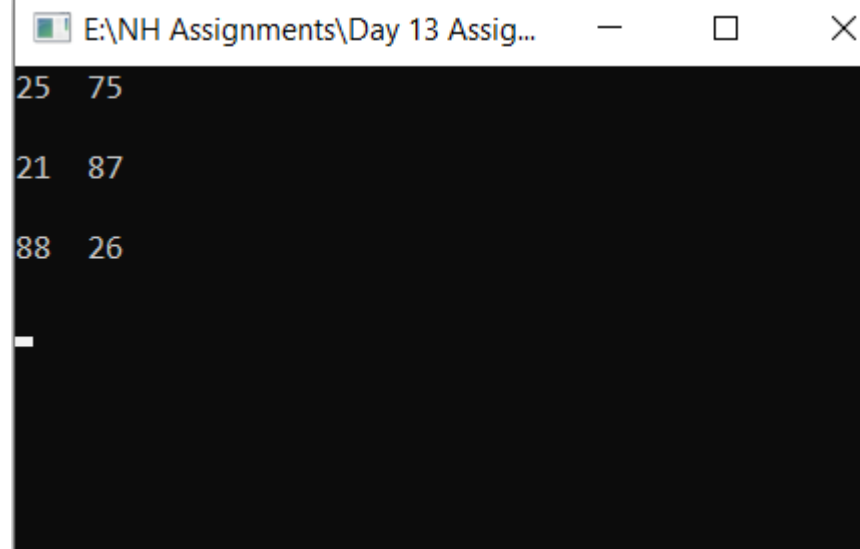
namespace Day_13_Project_2
{
    //Author: Mary Margaret
    //Two-Dimensional Array with size (3,2)

    internal class Program
    {
        static void Main(string[] args)
        {
            //2-D Array Initialization
            int[,] data = new int[3, 2] { { 25, 75 }, { 21, 87 }, { 88, 26 } };

            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:



```
E:\NH Assignments\Day 13 Assig...
25 75
21 87
88 26
```

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 Day_13_Project_3
{
    //Author: Mary Margaret
    //Trace of the matrix with size (3,3)
    internal class Program
    {
        static void Main(string[] args)
        {
            int i, j;
```


```
int sum = 0;
int[,] data = new int[3, 3] { { 25, 75, 44 }, { 21, 87, 29 }, { 88, 26, 71 } };

//To print matrix
for (i = 0; i < 3; i++)
{
    for (j = 0; j < 3; j++)
    {
        Console.Write(data[i,j] + " ");
    }
    Console.WriteLine("\n");
}

//To print Trace of matrix
for (i = 0; i < 3; i++)
{
    for (j = 0; j < 3; j++)
    {
        if (i == j)
            sum = sum + data[i, j];
    }
}
Console.WriteLine("Trace of the given matrix is : {0}", sum);

Console.ReadLine();
}
}
```

Output:



A screenshot of a console window titled "E:\NH Assignments\Day 13 Assignment by Mary Margarette o...". The console displays a 3x3 matrix of integers: 25, 75, 44 in the first row; 21, 87, 29 in the second row; and 88, 26, 71 in the third row. Below the matrix, it states "Trace of the given matrix is : 183".

```
25 75 44
21 87 29
88 26 71
Trace of the given matrix is : 183
```

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

Code:

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

namespace Day_13_Project_4
{
    //Author: Mary Margaret
    //2-D Array with user inputs
    internal class Program
    {
        static void Main(string[] args)
        {
            int i, j;

            int[,] data = new int[2, 2] ;
            for ( i = 0; i < 2; i++)
            {
                for ( j = 0; j < 2; j++)
                {
                    Console.WriteLine("Enter a number:");
                    data[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
        }
    }
}
```

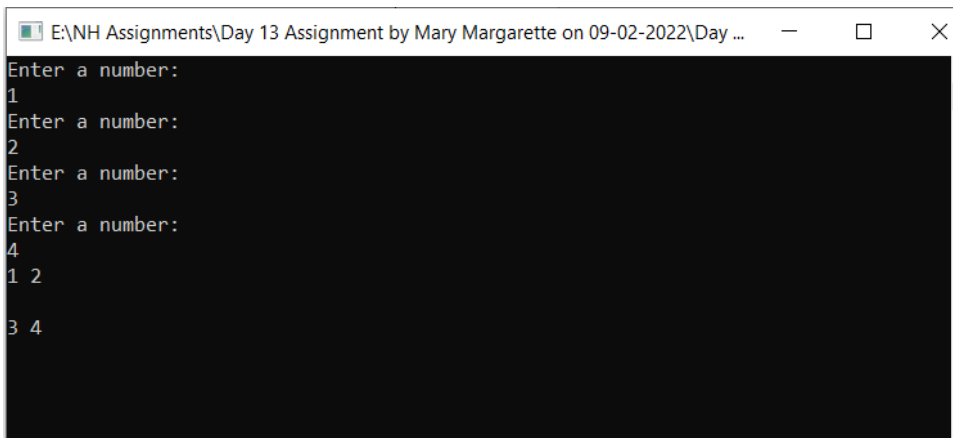
```

    }

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

```

Output:



```

E:\NH Assignments\Day 13 Assignment by Mary Margarette on 09-02-2022\Day ...
Enter a number:
1
Enter a number:
2
Enter a number:
3
Enter a number:
4
1 2
3 4

```

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

Code:

```

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

namespace Day_13_Project_10

```

```

{
    //Author: Mary Margaret
    //Adding of two Matrices
    internal class Program
    {
        static void Main(string[] args)
        {
            int i, j;
            int[,] data1 = new int[2, 2];
            int[,] data2 = new int[2, 2];
            int[,] data3 = new int[2, 2];

            //Matrix 1
            for (i = 0; i < 2; i++)
            {
                for (j = 0; j < 2; j++)
                {
                    Console.WriteLine("Enter a number:");
                    data1[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
            for (i = 0; i < 2; i++)
            {
                for (j = 0; j < 2; j++)
                {
                    Console.Write(data1[i, j] + " ");
                }
                Console.WriteLine("\n");
            }

            //Matrix 2
            for (i = 0; i < 2; i++)
            {
                for (j = 0; j < 2; j++)
                {
                    Console.WriteLine("Enter a number:");
                    data2[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
            for (i = 0; i < 2; i++)
            {
                for (j = 0; j < 2; j++)
                {
                    Console.Write(data2[i, j] + " ");
                }
                Console.WriteLine("\n");
            }
        }
    }
}

```

```
//Adding two matrices
```

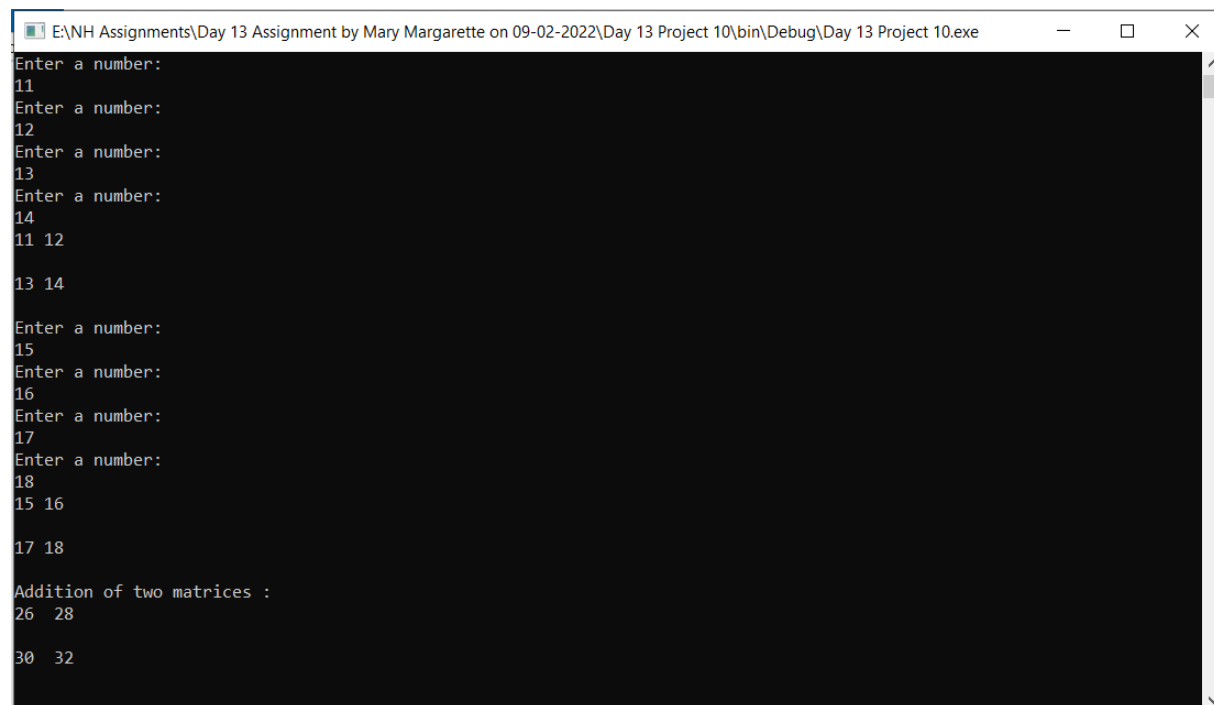
```
Console.WriteLine("Addition of two matrices :");
```

```
for (i = 0; i < 2; i++)  
{  
    for (j = 0; j < 2; j++)  
    {  
        data3[i, j] = data1[i, j] + data2[i, j];  
        Console.Write(data3[i, j] + " ");  
    }  
    Console.WriteLine("\n");  
}
```

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

```
    }  
}  
}
```

Output:



```
E:\NH Assignments\Day 13 Assignment by Mary Margarette on 09-02-2022\Day 13 Project 10\bin\Debug\Day 13 Project 10.exe  
Enter a number:  
11  
Enter a number:  
12  
Enter a number:  
13  
Enter a number:  
14  
11 12  
13 14  
Enter a number:  
15  
Enter a number:  
16  
Enter a number:  
17  
Enter a number:  
18  
15 16  
17 18  
Addition of two matrices :  
26 28  
30 32
```



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 Day_13_Project_11
{
    //Author: Mary Margaret
    //Multiplication of two matrices
    internal class Program
    {
        static void Main(string[] args)
        {
            //r1,c1 are rows and columns of matrix data1
            //r2,c2 are rows and columns of matrix data2
            int i, j;
            int r1 = 2;
            int r2 = 2;
            int c1 = 2;
            int c2 = 2;
            int[,] data1 = new int[r1, c1]; //array of data1
            int[,] data2 = new int[r2, c2]; //array of data2

            //Matrix 1 reading input
            for (i = 0; i < 2; i++)
            {
                for (j = 0; j < 2; j++)
                {
                    Console.WriteLine("Enter a number:");
                    data1[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
            //printing matrix data1

            for (i = 0; i < 2; i++)
            {
                for (j = 0; j < 2; j++)
                {
                    Console.Write(data1[i, j] + " ");
                }
                Console.WriteLine("\n");
            }
        }
    }
}
```

```

    }

    //Matrix 2 reading input
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 2; j++)
        {
            Console.WriteLine("Enter a number:");
            data2[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    //Printing matrix data2
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 2; j++)
        {
            Console.Write(data2[i, j] + " ");
        }
        Console.WriteLine("\n");
    }

    //Multiplication of two matrices
    if (c1 == r2) //condition for matrix multiplication
    {
        int[,] c = new int[r1, c2];
        Console.WriteLine("Matrix Multiplication:");
        for (i = 0; i < r1; i++)
        {
            for (j = 0; j < c2; j++)
            {
                c[i, j] = 0;
                for (int k = 0; k < r2; k++)
                {
                    c[i, j] += data1[i, k] * data2[k, j];
                }
                Console.Write(c[i, j] + " ");
            }
            Console.WriteLine("\n");
        }
        Console.ReadLine();
    }
}
}
}

```

Output:

```
E:\NH Assignments\Day 13 Assignment by Mary Margarette on 09-02-2022\Day 13 Project 11\bin\Debug\Day 13 Project 11.exe
Enter a number:
1
Enter a number:
2
Enter a number:
3
Enter a number:
4
1 2
3 4

Enter a number:
5
Enter a number:
6
Enter a number:
7
Enter a number:
8
5 6
7 8

Matrix Multiplication:
19 22
43 50
```

### What is a jagged array, What is the benefit of jagged array

- Jagged Array is a Two-Dimensional Array that can change size of the data dynamically.
- It is used to prevent Wastage of data.
- Syntax:  
`char [ ] [ ] names = new char [ 8] [ ];`
- Jagged Array saves data by changing size of array as per requirement.

### 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 Day_13_Project_5
{
    //Author: Mary Margaret
```

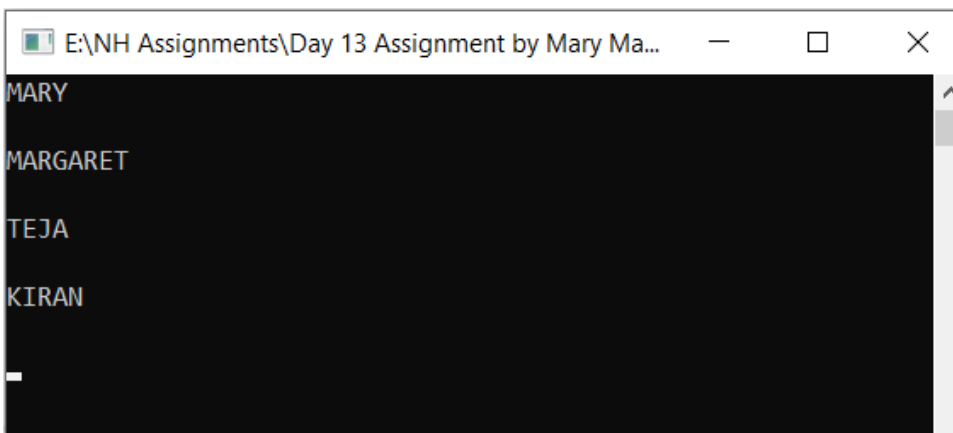
//Jagged Array

internal class Program

```
{
    static void Main(string[] args)
    {
        char[][] names = new char[4][];
        names[0] = new char[] { 'M', 'A', 'R', 'Y' };
        names[1] = new char[] { 'M', 'A', 'R', 'G', 'A', 'R', 'E', 'T' };
        names[2] = new char[] { 'T', 'E', 'J', 'A' };
        names[3] = new char[] { 'K', 'I', 'R', 'A', 'N' };

        for (int i = 0; i < 4; i++)
        {
            for (int j = 0; j < names[i].Length; j++)
            {
                Console.Write(names[i][j]);
            }
            Console.WriteLine("\n");
        }
        Console.ReadLine();
    }
}
```

Output:



## What is Recursion

- Function which calls itself until a specific condition is satisfied.
- Recursion is applied for Algorithms.
- Recursion is used for repeated calculations.

## WACP to illustrate usage of Recursion. What are the benefits of recursion

Code:

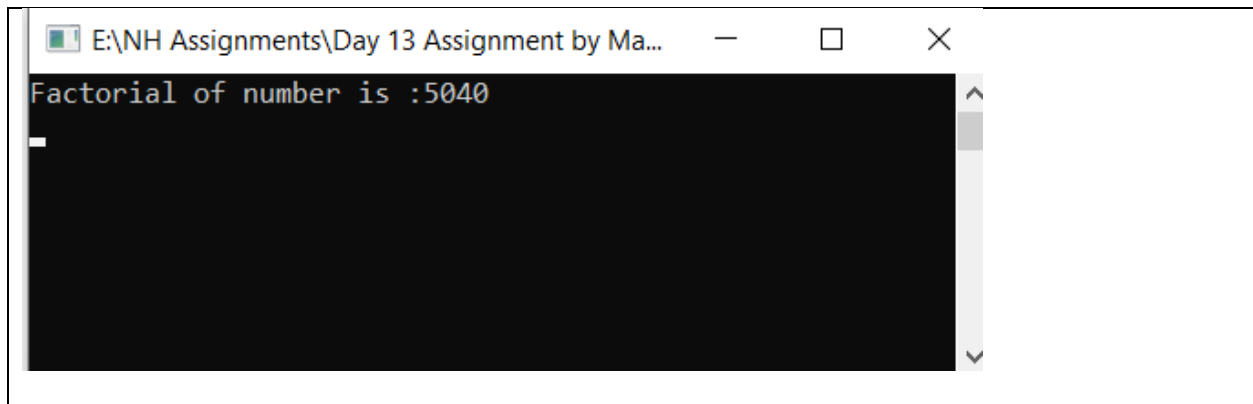
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day_13_Project_8
{
    //Author: Mary Margaret
    //Factorial with Recursion
    internal class Program
    {
        class Factorial
        {
            public static int Fact(int n)
            {
                if (n == 0)
                {
                    return 1;
                }

                else
                {
                    return n * Fact(n - 1);
                }
            }
        }

        static void Main(string[] args)
        {
            Console.WriteLine("Factorial of number is :{0}", Factorial.Fact(7));
            Console.ReadLine();
        }
    }
}
```

Output:



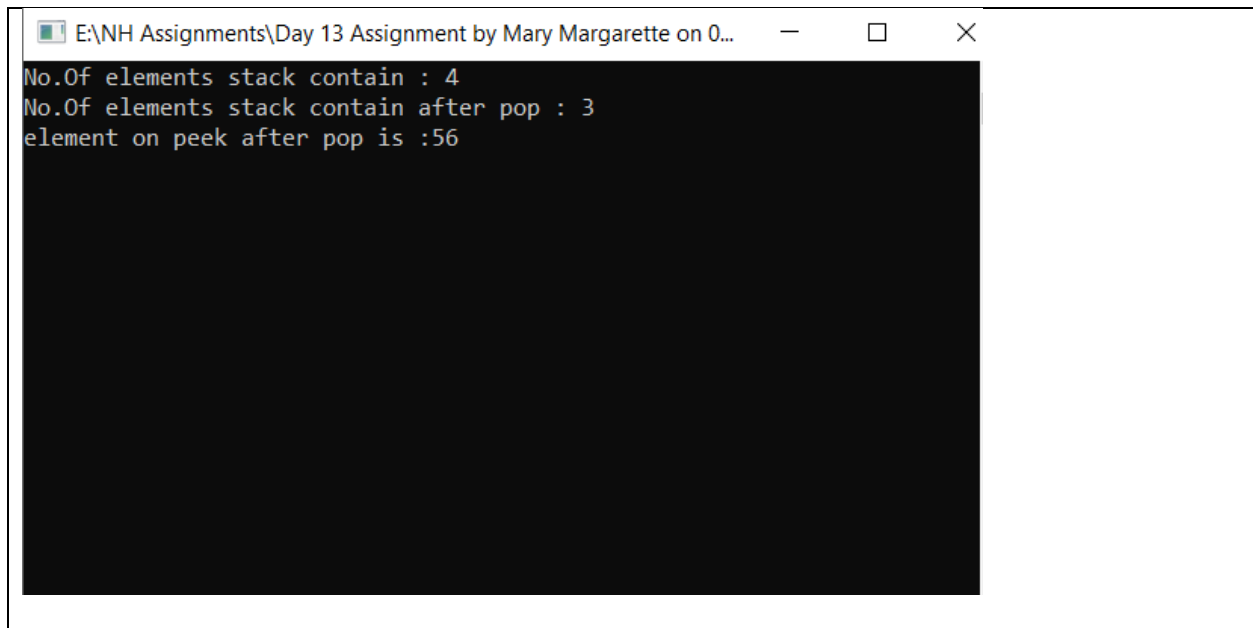
WACP to illustrate usage of Stack<>. Write couple of points about Stack

Code:

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

namespace Day_13_Project_6
{
    //Author: Mary Margaret
    //Stack Implementation
    internal class Program
    {
        static void Main(string[] args)
        {
            Stack<int> data = new Stack<int>();
            data.Push(18);
            data.Push(85);
            data.Push(56);
            data.Push(12);
            Console.WriteLine("No.Of elements stack contain : {0}", data.Count());
            data.Pop();
            Console.WriteLine("No.Of elements stack contain after pop : {0}", data.Count());
            Console.WriteLine("element on peek after pop is :{0}",data.Peek());
            Console.ReadLine();
        }
    }
}
```

Output:

A screenshot of a Windows console window. The title bar reads "E:\NH Assignments\Day 13 Assignment by Mary Margarette on 0...". The console output shows three lines of text: "No.Of elements stack contain : 4", "No.Of elements stack contain after pop : 3", and "element on peek after pop is :56". The rest of the console area is black.

```
E:\NH Assignments\Day 13 Assignment by Mary Margarette on 0...
No.Of elements stack contain : 4
No.Of elements stack contain after pop : 3
element on peek after pop is :56
```

WACP to illustrate usage of Queue<>. Write couple of points about Stack

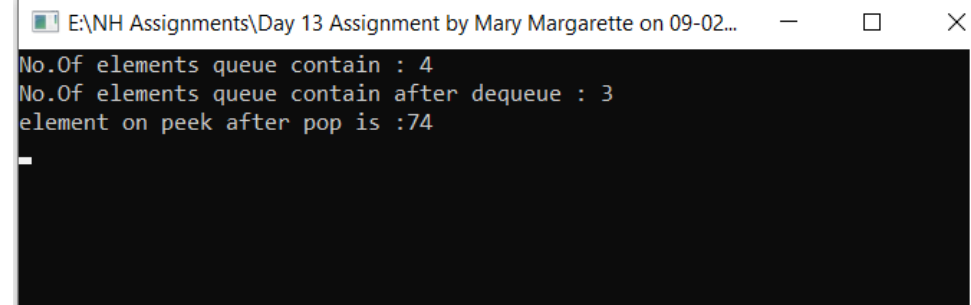
Code:

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

namespace Day_13_Project_7
{
    //Author: Mary Margaret
    //Queue Implementation
    internal class Program
    {
        static void Main(string[] args)
        {
            Queue<int> data = new Queue<int>();
            data.Enqueue(45);
            data.Enqueue(74);
            data.Enqueue(26);
            data.Enqueue(37);
            Console.WriteLine("No.Of elements queue contain : {0}", data.Count());
            data.Dequeue();
            Console.WriteLine("No.Of elements queue contain after dequeue : {0}", data.Count());
            Console.WriteLine("element on peek after pop is :{0}", data.Peek());
            Console.ReadLine();
        }
    }
}
```

```
}  
}  
}
```

#### Output:



The screenshot shows a Windows command prompt window with the title bar "E:\NH Assignments\Day 13 Assignment by Mary Margarete on 09-02...". The window contains the following text:

```
No.Of elements queue contain : 4  
No.Of elements queue contain after dequeue : 3  
element on peek after pop is :74
```