

**DAY 13 ASSIGNMENT**  
**BY**  
**NANAM VAISHNAVI**  
**09- FEB - 2022**

**1. 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;
// *****
// Author : Nanam Vaishnavi
// Purpose : 2D Array using nested for loop
// *****

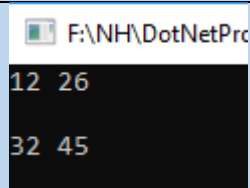
namespace Day13project1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data = new int[2, 2];
            {
                data[0, 0] = 12;
                data[0, 1] = 26;
                data[1, 0] = 32;
                data[1, 1] = 45;
            };

            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**



```
F:\NH\DotNetPro
12 26
32 45
```

**2. 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:**

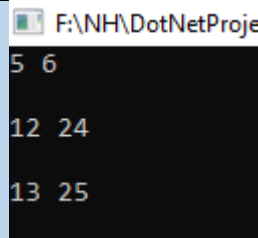
```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//
*****
// Author : Nanam Vaishnavi
// Purpose : initializing values in declaration line and printing values
//
/*****
namespace Day13Project2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data = new int[3, 2] { { 5, 6 }, { 12, 24 }, { 13, 25 } };
            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



```

F:\NH\DotNetProje
5 6
12 24
13 25

```

### 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;
// *****
// Author : Nanam Vaishnavi
// Purpose : Trace of the matrix
// *****/
namespace Day13Project3
{
    internal class Program
    {
        static void Main(string[] args)
        {

```

```

        int[,] data = new int[,] { { 1, 5, 9 }, { 2, 6, 10 }, { 3, 7, 11 } };
    };

    int sum = 0;
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (i == j)

                sum = sum + data[i, j];
        }
        Console.WriteLine("\n");
    }
    Console.WriteLine(sum);
    Console.ReadLine();
}
}
}

```

## OUTPUT

F:\NH\DotNetProjects\Day 13 Assignment\Day13Project3\Day13I  
18

**4. 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;
// *****
// Author : Nanam Vaishnavi
// Purpose : Read values from and print the matrix
// *****

namespace Day13Project4
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data = new int[2,2];

            // Read values from user

            for(int i =0;i<2;i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.WriteLine($"Enter array values at ({i},{j})");
                    data[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
            Console.WriteLine("\n");
            // Print Values
            for (int i = 0; i < 2; i++)

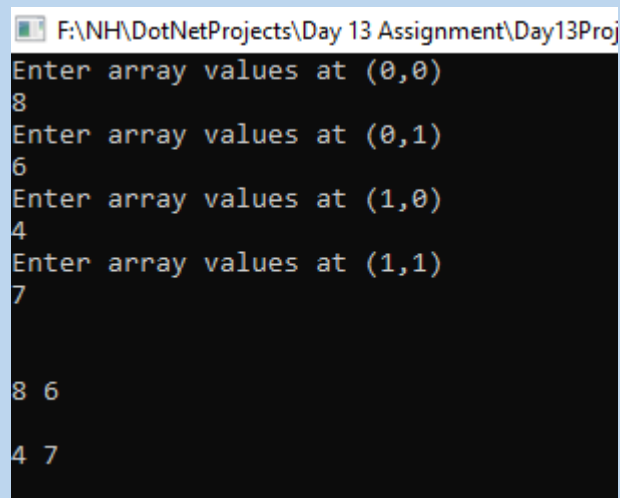
```

```

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

```

## OUTPUT



```

F:\NH\DotNetProjects\Day 13 Assignment\Day13Proj
Enter array values at (0,0)
8
Enter array values at (0,1)
6
Enter array values at (1,0)
4
Enter array values at (1,1)
7

8 6
4 7

```

**5. 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 Day13Project5
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] a1 = new int[2, 2];

            //Read values for arr1 from user

            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)

```

```

        {
            Console.WriteLine($"Enter First Matrix ({i},{j}): ");
            //Console.WriteLine(a1[i, j]);
            a1[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    int[,] a2 = new int[2, 2];

    //Read values for arr1 from user

    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            Console.WriteLine($" Enter Second Matrix({i},{j}): ");
            //Console.WriteLine(a2[i, j]);
            a2[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    // adding two matrixes
    Console.WriteLine("Addition of Two Matrices");

    int[,] a3 = new int[2, 2];
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            a3[i, j] = a1[i, j] + a2[i, j];
        }
    }

    // printing matrix
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            Console.Write(a3[i, j]+" ");
        }
        Console.WriteLine("\n \n");
    }
    Console.ReadLine();
}
}
}

```

**OUTPUT**

F:\NH\DotNetProjects\Day 13 Assignm

```
Enter First Matrix (0,0):
26
Enter First Matrix (0,1):
34
Enter First Matrix (1,0):
15
Enter First Matrix (1,1):
45
Enter Second Matrix(0,0):
16
Enter Second Matrix(0,1):
18
Enter Second Matrix(1,0):
29
Enter Second Matrix(1,1):
35
Addition of Two Matrices
42 52

44 80
```

**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;
// *****
// Author : Nanam Vaishnavi
// Purpose : Product of two Matrices.
//*****

namespace Day13Project6
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] p1 = new int[2, 2];

            //Read values for p1 from user

            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.WriteLine($"Enter First Matrix ({i},{j}): ");
                    p1[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }
        }
    }
}
```

```

    }

    int[, ] p2 = new int[2, 2];

    //Read values for p2 from user
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            Console.WriteLine($" Enter Second Matrix({i},{j}): ");

            p2[i, j] = Convert.ToInt32(Console.ReadLine());
        }
    }

    // Product of two matrixes
    Console.WriteLine("Product of Two Matrices");

    int[, ] p3 = new int[2, 2];
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            p3[i, j] = p1[i, j] * p2[i, j];
        }
    }

    // printing matrix
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < 2; j++)
        {
            Console.Write(p3[i, j] + " ");
        }
        Console.WriteLine("\n");
    }
    Console.ReadLine();
}
}
}

```

**OUTPUT**



```

F:\NH\DotNetProjects\Day 13 Assignm
Enter First Matrix (0,0):
5
Enter First Matrix (0,1):
3
Enter First Matrix (1,0):
9
Enter First Matrix (1,1):
4
Enter Second Matrix(0,0):
8
Enter Second Matrix(0,1):
6
Enter Second Matrix(1,0):
7
Enter Second Matrix(1,1):
1
Product of Two Matrices
40 18
63 4

```

## 7. What is a jagged Array. What is the benefit of jagged Array.

**Ans :**

- A Jagged array is an array of arrays such that member arrays can be of different sizes.
- Each of the elements is a single – dimensional array of integers.

### **Benefits of Jagged Array:**

- It makes things easy where there is a need to store data in a multidimensional way using the same variable name.
- No need to mention column size.

## 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;
// *****
// Author : Nanam Vaishnavi
// Purpose : Declare Jagged Array and print values
// *****

namespace Day13project7
{
    internal class Program
    {
        static void Main(string[] args)
        {
            char[][] names = new char[3][];

```

```

names[0] = new char[] { 'A', 'b', 'h', 'i' };
names[1] = new char[] { 'R', 'a', 't', 'h', 'i', 'k', 'a' };
names[2] = new char[] { 'K', 'a', 'r', 't', 'h', 'i', 'k' };

for(int i=0;i<3;i++)
{
    for(int j =0;j<names[i].Length;j++)
    {
        Console.Write(names[i][j]);
    }
    Console.WriteLine("\n");
}
Console.ReadLine();
}
}

```

#### OUTPUT

F:\NH\DotNetProjects\Day 13 Assignment\Day13project7\Day13project7\

```

Abhi
Rathika
Karthik

```

#### 9. What is Recursion ?

Ans : Recursion is defined as function calling itself until a specified condition is satisfied.

#### 10. Write a C# program to illustrate usage of Recursion. What are the benefits of recursion

##### Benefits of Recursion:

1. It adds clarity and reduces the time needed to write and debug.
2. Reduces time complexity.

##### CODE

##### OUTPUT:

#### 11. Write a C# Program to illustrate usage of Stack<>. Write couple of points about Stack

##### Stack :

- A stack is an abstract data type which holds an ordered, sequence of items.
- Stack uses **Last In First Out (LIFO)** structure.

##### CODE

```

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

```

```
// *****
// Author : Nanam Vaishnavi
// Purpose : Usage of Stack
// *****

namespace Day13Project9
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Stack<int> data = new Stack<int>();
            {
                data.Push(26);
                data.Push(20);
                data.Push(37);
                Console.WriteLine(data.Count);
                Console.WriteLine(data.Pop());
                Console.WriteLine(data.Count);
                Console.WriteLine(data.Pop());

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

## OUTPUT

F:\NH\DotNetProjects\Day 13 Assignment\Day13Project9\Day13Project9\bin\Debug\l

```
3
37
2
20
```

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

### Queue :

- ❖ Queue is used when things don't have to be processed immediately.
- ❖ It follows **First In First Out (FIFO)** order.

### CODE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
// *****
// Author : Nanam Vaishnavi
// Purpose : Usage of Queue
// *****


namespace Day13Project10
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Queue<int> data = new Queue<int>();
```

```
data.Enqueue(56);  
data.Enqueue(45);  
data.Enqueue(50);  
Console.WriteLine(data.Count);  
Console.WriteLine(data.Dequeue());
```

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

```
    }  
}
```

## OUTPUT

 F:\NH\DotNetProjects\Day 13 Assignment\Day13Project10\Day13Project10\bin\Debug\Day13Project10.exe

3

56