

Day 13(09-02-2022) Assignment

By

Sudha Kumari Sugasani

Q1.Declare a 2 dimensional array of size (2,2) and initialize using indexes and print the values using nested for loop.

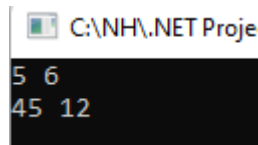
Code:

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

namespace Day13Project1
{
    /*****
     * Author:Sudha Kumari Sugasani
     * Purpose: Declaration and intialisation of 2 dimensional array
     *           of size(2x2),print the values using nested for loop
     * *****/
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data = new int[2, 2];
            data[0, 0] = 5;
            data[0, 1] = 6;
            data[1, 0] = 45;
            data[1, 1] = 12;

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

Output:



```
C:\NH\..NET Proje
5 6
45 12
```

Q2.Declare a 2 dimensional 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;
```

```

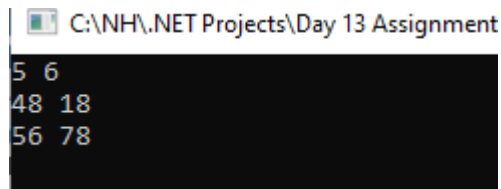
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Day13Project2
{
    /*****
    * Author:Sudha Kumari Sugasani
    * Purpose: Declaration and intialisation of 2 dimensional array
    *          in same line of size(3x2),print the values using
    *          nested for loop
    * *****/
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data = new int[,] { { 5, 6 }, { 48, 18 }, { 56, 78 } };

            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.Write(data[i, j] + " ");
                }
                Console.WriteLine();
            }
            Console.ReadLine();
        }
    }
}

```

Output:



```

C:\NH\NET Projects\Day 13 Assignment
5 6
48 18
56 78

```

Q3.Declare a 2 dimensional array of size (3x3) 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:Sudha Kumari Sugasani
    * Purpose: Declaration of 2 dimensional array of size(3x2),
    *          print trace of the array.
    * *****/
    internal class Program
    {
        static void Main(string[] args)

```

```

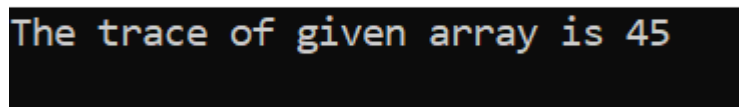
    {
        int sum = 0;
        int[,] data = new int[,] { { 5, 6 ,15}, { 48, 18 ,13}, { 56,
78,22 } };

        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                if(i==j)
                {
                    sum=sum+data[i,j];
                }
            }
        }
        Console.WriteLine($"The trace of given array is {sum}");
        Console.ReadLine();
    }
}
}
}

```

Output:

C:\NH\NET Projects\Day 13 Assignment\DayProject3\DayProje



The trace of given array is 45

Q4.Declare a 2 dimensional array of size (2x2) and read values from the user and print the array values.

Code:

```

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

namespace Day13Project4
{
    /*****
    * Author:Sudha Kumari Sugasani
    * Purpose: Declaration of 2 dimensional array of size(2x2),
    *          read values from user and print the array values.
    * *****/
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data = new int[2, 2];
            //Read data from user

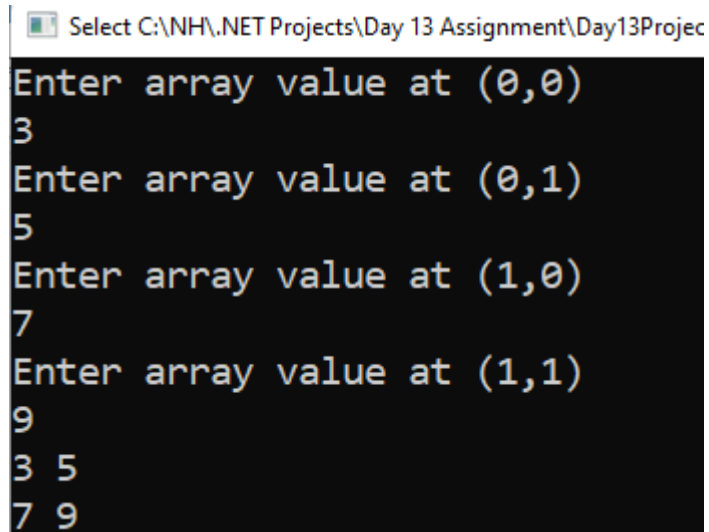
```

```

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

```

Output:



```

Select C:\NH\NET Projects\Day 13 Assignment\Day13Project
Enter array value at (0,0)
3
Enter array value at (0,1)
5
Enter array value at (1,0)
7
Enter array value at (1,1)
9
3 5
7 9

```

Q5.Declare two 2 dimensional arrays of size(2,2) and read values from user and print the sum of two matrices.

Code:

```

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

namespace Day13project5
{
    /*****
    * Author:Sudha Kumari Sugasani

```

```

* Purpose:Sum of two 2 dimensional arrays of size(2x2)
* *****/
internal class Program
{
    static void Main(string[] args)
    {
        int [,] data1=new int[2,2];
        //Reading array1 values from user
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                Console.WriteLine($"Enter value at data1 ({i},{j})");
                data1[i,j]=Convert.ToInt32(Console.ReadLine());
            }
        }

        int[,] data2= new int[2, 2];
        //Reading array2 values from user
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 2; j++)
            {
                Console.WriteLine($"Enter value at data2 ({i},{j})");
                data2[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }
        //Printing addition of two arrays
        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                Console.Write(data1[i,j]+data2[i,j]+" ");
            }
            Console.WriteLine();
        }
        Console.ReadLine();
    }
}

```

Output:

C:\NH\NET Projects\Day13project5\Day13project5\bin\Debug

```
Enter value at data1 (0,0)
1
Enter value at data1 (0,1)
2
Enter value at data1 (1,0)
3
Enter value at data1 (1,1)
4
Enter value at data2 (0,0)
5
Enter value at data2 (0,1)
6
Enter value at data2 (1,0)
7
Enter value at data2 (1,1)
8
6 8
10 12
```

Q6. Declare two 2 dimensional arrays of size(2,2) and read values from user and print the product of two matrices.

Code:

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

namespace Day13Project6
{
    /******
    * Author:Sudha Kumari Sugasani
    * Purpose:Product of two 2 dimensional arrays of size(2x2)
    * *****/
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data1 = new int[2, 2];
            //Reading array1 values from user
            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.WriteLine($"Enter value at data1 ({i},{j})");
                    data1[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }

            int[,] data2 = new int[2, 2];
            //Reading array2 values from user
            for (int i = 0; i < 2; i++)
```

```

        {
            for (int j = 0; j < 2; j++)
            {
                Console.WriteLine($"Enter value at data2 ({i},{j})");
                data2[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }
        //Printing addition of two arrays
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 2; j++)
            {
                Console.Write(data1[i, j] * data2[i, j] + " ");
            }
            Console.WriteLine();
        }
        Console.ReadLine();
    }
}

```

Output:

```

C:\NH\...NET Projects\Day13Project6\Day13Project6\bin\Debug\Day13Pr
Enter value at data1 (0,0)
1
Enter value at data1 (0,1)
2
Enter value at data1 (1,0)
3
Enter value at data1 (1,1)
4
Enter value at data2 (0,0)
5
Enter value at data2 (0,1)
6
Enter value at data2 (1,0)
7
Enter value at data2 (1,1)
8
5 12
21 32

```

Q7.What is a jagged array

What is the benefit of jagged array

- It is a 2 dimensional array.
- It helps in memory management which makes the program to be executed smooth and fast
- In this each row of different size whereas in normal arrays each row of same size

Q8.Write a C# program to declare a jagged array and print the values.(using int)

Code:

```
using System;
```

```

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

namespace Day13project7
{
    /**
     * Author:Sudha Kumari Sugasani
     * Purpose:Declaration and printing of a jagged array
     * ***/
    internal class Program
    {
        static void Main(string[] args)
        {
            //Declaration and intialisation of jagged array
            int [][] values=new int[10][];
            values[0] = new int[] {15,1,2};
            values[1] = new int[] { 25 ,3,2,1};
            values[2] = new int[] { 33 ,55,66,77,1234};
            values[3] = new int[] { 5 ,7689,789098,2231};
            values[4] = new int[] { 10,67,89,22 ,12,11};
            values[5] = new int[] { 18 ,15,14,13,5};
            values[6] = new int[] { 46 ,55,45,35};
            values[7] = new int[] { 7,789,2223};
            values[8] = new int[] { 2,5,7,9};
            values[9] = new int[] { 11 };

            //Printing of jagged array values
            for(int i = 0; i < 10; i++)
            {
                for (int j = 0;j<values[i].Length;j++)
                {
                    Console.Write(values[i][j]+" ");
                }
                Console.WriteLine("\n");
            }
            Console.ReadLine();
        }
    }
}

```

Output:

C:\NH\..NET Projects\Day13project7\

```
15 1 2
25 3 2 1
33 55 66 77 1234
5 7689 789098 2231
10 67 89 22 12 11
18 15 14 13 5
46 55 45 35
7 789 2223
2 5 7 9
11
```

Q8. Write a C# program to declare a jagged array and print the values (using character array)

Code:

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

namespace Day13project8
{
    /*****
     * Author: Sudha Kumari Sugasani
     * Purpose: Declaration and printing of a jagged array
     *           using character array
     * *****/
    internal class Program
    {
        static void Main(string[] args)
        {
            //Declaration and initialisation of jagged character array
            char[][] data = new char[5][];
            data[0] = new char[] { 's', 'u', 'd', 'h', 'a' };
            data[1] = new char[] { 's', 'u', 'd', 'h', 'a', 's' };
            data[2] = new char[] { 's', 'u', 'd', 'h', 'a', 's', 'u' };
            data[3] = new char[] { 's', 'u', 'd', 'h', 'a', 's', 'u', 'g' };
            data[4] = new char[] { 's', 'u', 'd', 'h',
            'a', 's', 'u', 'g', 'a', 's', 'a', 'n', 'i' };

            //Printing of jagged character array values
            for (int i = 0; i < 5; i++)
            {
                for (int j = 0; j < data[i].Length; j++)
                {
                    Console.Write(data[i][j] + " ");
                }
                Console.WriteLine("\n");
            }
            Console.ReadLine();
        }
    }
}
```

```

    }
}
}

```

Output:

C:\NH\NET Projects\Day13project8\Day13proj

```

s u d h a
s u d h a s
s u d h a s u
s u d h a s u g
s u d h a s u g a s a n i

```

Q9.What is Recursion?

A function calling itself repeatedly until a specified condition is satisfied is called recursion.

Q10.Write a C# program to illustrate usage of Recursion?

What are the benefits of recursion?

Benefits:

- 1.It reduces memory by using less variables.
- 2.It reduces duplicate code because in this recursion we are calling the same method repeatedly until a specified condition satisfies.

Code:

```

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

namespace Day10Project9
{
    /*****
     * Author:Sudha Kumari Sugasani
     * Purpose:Example program for recursion
     * *****/
    internal class Program
    {
        /// <summary>
        /// It will return factorial of a number
        /// </summary>
        /// <param name="n">5</param>
        /// <returns>int</returns>
        static int Factorial(int n)
        {
            if(n== 0)
                return 1;
            else
                return n*Factorial(n-1);
        }
        static void Main(string[] args)
        {
            Console.WriteLine($"Factorial of given number is {Factorial(5)}"); ;
            Console.ReadLine();
        }
    }
}


```

```

    }
}
}

```

Output:

 C:\NH\NET Projects\Day10Project9\Day10Project9\bin\Debug\Day10Project9.e

Factorial of given number is 120

Q11. Write a C# program to illustrate usage of stack
Write couple of points about stack.

Points about stack:

- In this we use Push() to add elements.
- In this we use Pop() to remove elements.
- It follows last in first out process.
- Peek() is used to return the top element in the stack.

Code:

```

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

namespace Day13Project10
{
    /*****
     * Author:Sudha Kumari Sugasani
     * Purpose:Example program for stack
     * *****/
    internal class Program
    {
        static void Main(string[] args)
        {
            Stack<int> data = new Stack<int>();
            data.Push(10);
            data.Push(20);
            data.Push(30);
            Console.WriteLine($"Count before Pop is {data.Count}");
            Console.WriteLine($"Pop value is {data.Pop()}");
            Console.WriteLine($"Count after Pop is {data.Count}");
            Console.WriteLine($"Count before Peek is {data.Count}");
            Console.WriteLine($"Peek value is {data.Peek()}");
            Console.WriteLine($"Count after Peek is {data.Count}");
            Console.ReadLine();
        }
    }
}

```

Output:

C:\NH\ .NET Projects\Day13Project10\Day1

```
Count before pop is 3
Pop value is 30
Count after pop is 2
Count before peek is 2
Peek value is 20
Count after peek is 2
```

Q12. Write a C# program to illustrate the usage of Queue.
Write couple of points about Queue

Points about Queue:

- In this we use Enqueue() to add elements.
- In this we use Dequeue() to remove elements.
- It follows first in first out process.
- Peek() is used to return the first element in the Queue.

Code:

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

namespace Day13Project11
{
    /*****
     * Author: Sudha Kumari Sugasani
     * Purpose: Example program for Queue
     * *****/
    internal class Program
    {
        static void Main(string[] args)
        {
            Queue<int> data = new Queue<int>();
            data.Enqueue(10);
            data.Enqueue(15);
            data.Enqueue(30);
            Console.WriteLine($"Count before Dequeue is {data.Count}");
            Console.WriteLine($"Dequeue value is {data.Dequeue()}");
            Console.WriteLine($"Count after Dequeue is {data.Count}");
            Console.WriteLine($"Count before Peek is {data.Count}");
            Console.WriteLine($"Peek value is {data.Peek()}");
            Console.WriteLine($"Count after Peek is {data.Count}");
            Console.ReadLine();
        }
    }
}
```

Output:

C:\NH\NET Projects\Day13Project11\Day13Project11\

```
Count before Dequeue is 3  
Dequeue value is 10  
Count after Dequeue is 2  
Count before Peek is 2  
Peek value is 15  
Count after Peek is 2  
_
```