Day 13 Assignment
By
Triveni Anumolu
09-02-2022

1.Declare a 2 dimentional 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
{
   class Program
   {
       /*****************************************************************
         Author: Triveni Anumolu
         Purpose: Printing 2 dimensional array using nested for loop
       *****************************************************************/
      static void Main(string[] args)
      {

         int[,] data = new int[2, 2];
         data[0, 0] = 1;
         data[0, 1] = 2;
         data[1, 0] = 3;
         data[1, 1] = 4;
         for(int i=0;i<2;i++)
         {
            for(int j=0; j<2; j++)
            {
               Console.Write(data[i,j] + " ");
            }
            Console.WriteLine();
         }
         Console.ReadLine();
```

```
        }
    }
}
```

Result:

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;

namespace Day13project2
{
    class Program
    {
        /*************************************************************
        Author: Triveni Anumolu
        Purpose: Declaring a 2-D array  and initializing in the same line
        *************************************************************/
        static void Main(string[] args)
        {

            int[,] data = new int[ , ] { { 1, 2 } ,{ 4, 5 }, { 8, 9 } };


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

        }
    }
```

```
}
```

Result:

```
1 2
4 5
8 9
```

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
{
   class Program
   {
      /*************************************************************
       Author:Triveni Anumolu
        Purpose: Printing the trace of array
       *************************************************************/
      static void Main(string[] args)
      {
         int[,] data = new int[,] { { 1, 2 ,1}, { 4, 5,9 }, { 8, 9,7 } };
         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("Trace of array is " + sum);
         Console.ReadLine();
      }
   }
}
```

Result:

```
Trace of array is 13
```

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;

namespace Day13project4
{
   class Program
   {
       /*************************************************************
        Author: Triveni Anumolu
        Purpose: Reading values from user and Printing values in a 2-D array
       *************************************************************/
      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 a number at ({i},{j})");
               data[i, j] = Convert.ToInt32(Console.ReadLine());
            }
         }
         for (int i = 0; i < 2; i++)
         {
            for (int j = 0; j < 2; j++)
            {
               Console.Write(data[i, j] + " ");
            }
            Console.WriteLine();
         }
         Console.ReadLine();

      }
   }
}
```

Result:

```
Enter a number at (0,0)
12
Enter a number at (0,1)
45
Enter a number at (1,0)
32
Enter a number at (1,1)
21
12 45
32 21
```

5.Declare TWO 2-D arrays of size (2,2) and read values fromuser 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 Day13project9
{
   class Program
   {
      /*************************************************************
       Author: Triveni Anumolu
       Purpose: sum of  two matrices
      *************************************************************/
      static void Main(string[] args)
      {
        int[,] data1 = new int[2, 2];
          int[,] data2 = new int[2, 2];
          int[,] sum = new int[2, 2];
          //Read data from user for first matrix

          for (int i = 0; i < 2; i++)
          {
             for (int j = 0; j < 2; j++)
             {
                Console.WriteLine($"Enter a number for first matrix at ({i},{j})");
                data1[i, j] = Convert.ToInt32(Console.ReadLine());
             }
          }
```

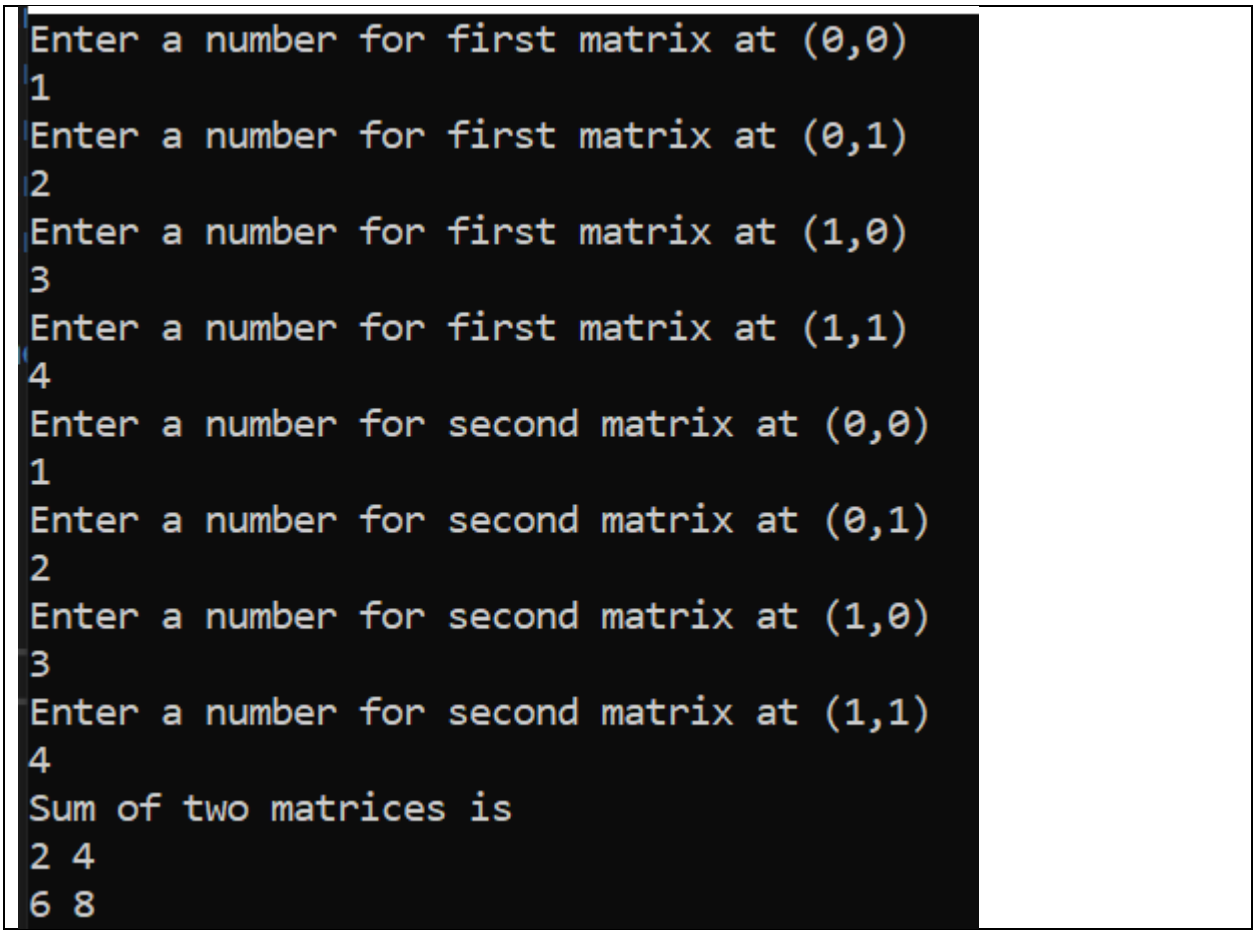
```
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 2; j++)
            {
                Console.WriteLine($"Enter a number for second matrix at ({i},{j})");
                data2[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }
        Console.WriteLine("Sum of two matrices is");

        for(int i=0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                sum[i, j] = data1[i, j] + data2[i, j];
                Console.Write(sum[i,j] +" ");

            }
            Console.WriteLine();
        }
        Console.ReadLine();

    }
  }
}
```

Result:

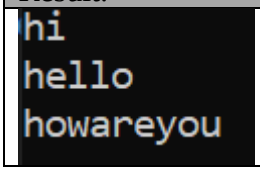


7. What is a jagged array What is the benefit of jagged array
Jagged Array:
Jagged Array is a 2-Dimensional array which has different sizes for different rows.
Benefit of Jagged Array:
It helps in memory management which makes the program to be executed very smoothly and fast as well.

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 Day13Project5

```
{
   class Program
   {
      /*************************************************************
       Author: Triveni Anumolu
      Purpose:printing  elements of a jagged array
      *************************************************************/
      static void Main(string[] args)
      {
         char[][] ch = new char[3][];
         ch[0] = new char[] { 'h', 'i' };
         ch[1] = new char[] { 'h', 'e' ,'l','l','o'};
         ch[2] = new char[] { 'h', 'o','w','a','r','e','y','o','u'};
         for(int i=0;i<3;i++)
         {
            for(int j=0;j<ch[i].Length;j++)
            {
               Console.Write(ch[i][j]);
            }
            Console.WriteLine();
         }
         Console.ReadLine();
      }
   }
}
```

Result:



9.What is Recursion.

A. Recursion means a function calling itself repeatedly until a specific condition is satisfied.

10. 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 Day13project6
```

```
{

   class Program
   {
      /*************************************************************
       Author: Triveni Anumolu
       Purpose: Finding factorial of a number using recursion
      *************************************************************/
      public static int Factorial(int n)
      {
         if (n == 0)
            return 1;
         else
            return n * Factorial(n - 1);
      }
      public static void Print(int n)
      {
         Console.WriteLine("Factorial of {0} is {1}", n, Factorial(n));
      }
      static void Main(string[] args)
      {
         int n = 9;
         Print(n);
         Console.ReadLine();
      }
   }
}
```

Result:

```
Factorial of 9 is 362880
```

Benefits of Recursion:

- For a recursive function, you only need to define the base case and recursive case, so the code is simpler.
- Recursion reduces the length of code.
- Reduces time complexity.

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

Stack:

- Stack is one of the types in Generics.
- Stack follows Last In First Out(LIFO) order.
- Push() is used to add the data in stack.
- Pop() deletes the elements from stack.

Code:

```
using System;
using System.Collections.Generic;
```

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

namespace Day13project7
{
   class Program
   {
      static void Main(string[] args)
      {
         /*********************************************************
           Author:Triveni Anumolu
           Purpose:performing operations on stack
         *********************************************************/

         Stack<int> data = new Stack<int>();
         data.Push(6);
         data.Push(1);
         data.Push(7);
         data.Push(76);
         Console.WriteLine(data.Count);
         Console.WriteLine(data.Peek());
         Console.WriteLine(data.Pop());
         Console.WriteLine(data.Count());
         Console.ReadLine();
      }
   }
}
```

Result:

```
4
76
76
3
```

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

Queue:

- Queue is one of the types in Generics.
- Queue follows First In First Out(FIFO) order.
- Enqueue() is used to add the elements in a queue.
- Dequeue() deletes the elements in a queue.

Code:

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

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

namespace Day13project8
{
   class Program
   {
      static void Main(string[] args)
      {
         /*************************************************************
           Author:Triveni Anumolu
           Purpose:performing operations on queue
         *************************************************************/

         Queue<int> data = new Queue<int>();
         data.Enqueue(6);
         data.Enqueue(1);
         data.Enqueue(7);
         data.Enqueue(76);
         Console.WriteLine(data.Count);
         Console.WriteLine(data.Peek());
         Console.WriteLine(data.Dequeue());
         Console.WriteLine(data.Count());
         Console.ReadLine();
      }
   }
}
```

Result: