



DAY13 ASSIGNMENTS

Bhanu Prakash Reddy Chilukuri

NB HEALTHCARE TECHNOLOGIES

09-02-2022





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

Code:

```
//Author: Bhanu Prakash Reddy
//WACP for 2D Array of size (2,2) using nested for loop
internal class Program
{
    static void Main(string[] args)
    {
        int[,] data = new int[,]
        {
            {5,8},
            {16,14}
        };
        //nested for loop
        for (int i = 0; i < 2; i++)
        {
            for(int j = 0; j < 2; j++)
            {
                Console.Write(data[i,j] + " ");
            }
            Console.WriteLine();
        }
        Console.ReadLine();
    }
}
```

Output:

 D:\assignments\Day13Project1\Day13Project1\bin\Debug\Day13Project1.exe



```
5 8
16 14
```

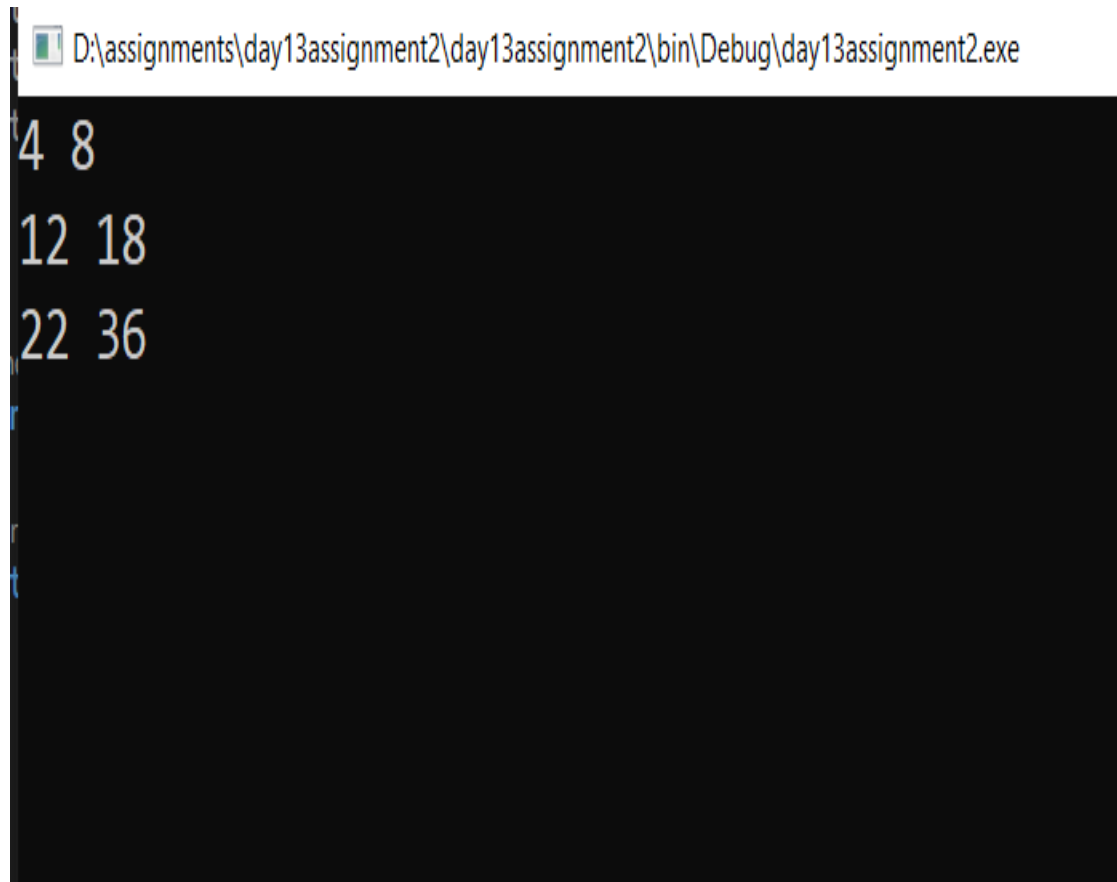
2. Declare a 2-D array of size (3,2) and initialize in the same line while declaring and print the values using nested for loop.

Code:

```
//Author: Bhanu Prakash Reddy
//WACP for 2D array of size (3,2) and initialize in same line and print using
nested for loop
internal class Program
{
    static void Main(string[] args)
    {
        int[,] data = { { 4, 8 }, { 12, 18 }, { 22, 36 } };

        //Nested for loop
        for(int i = 0; i < data.GetLength(0); i++)
        {
            for (int j = 0; j < data.GetLength(1); j++)
            {
                Console.Write(data[i, j] + " ");
            }
            Console.WriteLine("\n");
        }
        Console.ReadLine();
    }
}
```

Output:



```
D:\assignments\day13assignment2\day13assignment2\bin\Debug\day13assignment2.exe
4 8
12 18
22 36
```


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

Code:

```
//Author: Bhanu Prakash Reddy
// WACP for finding the trace of the array of size (3,3)
internal class Program
{
    static void Main(string[] args)
    {
        int[,] data = { { 4, 8, 9 }, { 15, 23, 67 }, { 14, 32, 18 } };
        int sum = 0;

        for (int i = 0; i < data.GetLength(0); i++)
        {
            for (int j = 0; j < data.GetLength(1); j++)
            {
                //Trace of the Matrix
                if (i == j)
                {
                    sum += data[i, j];
                }
            }
        }
        Console.WriteLine($"Trace of the matrix is {sum}");
        Console.ReadLine();
    }
}
```

Output:

 D:\assignments\Day13 Assignment3\Day13 Assignment3\bin\Debug\Day13 Assignment3.exe

Trace of the matrix is 45

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

Code:

```
// Author: Bhanu Prakash Reddy
//WACP 2D array of size (2,2) and read values from user and print values
internal class Program
{
    static void Main(string[] args)
    {
        int[,] data = new int[2, 2];
        //Read Values from user
        for(int i=0; i<data.GetLength(0); i++)
        {
            for(int j=0; j<data.GetLength(1); j++)
            {
                Console.WriteLine($"Enter the value at ({i},{j}) : ");
                data[i,j] = Convert.ToInt32(Console.ReadLine());
            }
        }
        //Print values
        for (int i = 0; i < data.GetLength(0); i++)
        {
            for (int j = 0; j < data.GetLength(1); j++)
            {
                Console.Write(data[i,j]+" ");
            }
            Console.WriteLine("\n");
        }
        Console.ReadLine();
    }
}
```

Output:

Select D:\assignments\Day13 assignment4\Day13 assignment4\bin\Debug\Day13 assignment4.exe

```
Enter the value at (0,0) :
14
Enter the value at (0,1) :
65
Enter the value at (1,0) :
45
Enter the value at (1,1) :
32
14 65
45 32
```

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

Code:

```
//Author : Bhanu Prakash Reddy
//WACP sum of the matrices
internal class Program
{
    static void Main(string[] args)
    {
        int[,] data1 = new int[2, 2];
        int[,] data2 = new int[2, 2];
        int[,] data3 = new int[2, 2];
        //Read Values from user
        for (int i = 0; i < data1.GetLength(0); i++)
        {
            for (int j = 0; j < data1.GetLength(1); j++)
            {
                Console.WriteLine($"Enter the value at ({i},{j}) : ");
                data1[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }
        //print data1 array
        for (int i = 0; i < data1.GetLength(0); i++)
        {
            for (int j = 0; j < data1.GetLength(1); j++)
            {
                Console.Write(data1[i, j] + " ");
            }
            Console.WriteLine("\n");
        }

        //read value from user
        for (int i = 0; i < data2.GetLength(0); i++)
        {
            for (int j = 0; j < data2.GetLength(1); j++)
            {
                Console.WriteLine($"Enter the value at ({i},{j}) : ");
                data2[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }
        //print data2 array
        for (int i = 0; i < data2.GetLength(0); i++)
        {
            for (int j = 0; j < data2.GetLength(1); j++)
            {
                Console.Write(data2[i, j] + " ");
            }
            Console.WriteLine("\n");
        }


        //sum of the two matrices
        for (int i = 0; i < data3.GetLength(0); i++)
        {
            for (int j = 0; j < data3.GetLength(1); j++)
            {
                int sum = data1[i, j] + data2[i, j];
                data3[i, j] = sum;
            }
        }
        Console.WriteLine($"Sum of tho Matrices :");
    }
}
```

```

        for (int i = 0; i < data3.GetLength(0); i++)
        {
            for (int j = 0; j < data3.GetLength(1); j++)
            {
                Console.Write($"{data3[i, j]} ");
            }
            Console.WriteLine("\n");
        }
        Console.ReadLine();
    }
}

```

Output:

 D:\assignments\DAy13Assignment5\DAy13Assignment5\bin\Debug\DAy13Assignment5.exe

```

Enter the value at (0,0) :
11
Enter the value at (0,1) :
12
Enter the value at (1,0) :
13
Enter the value at (1,1) :
14
11 12
13 14
Enter the value at (0,0) :
15
Enter the value at (0,1) :
16
Enter the value at (1,0) :
17
Enter the value at (1,1) :
18
15 16
17 18
Sum of tho Matrices :
26 28
30 32

```

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

Code:

```
//Author: Bhanu Prakash Reddy
// WACP for the product of two matrices
internal class Program
{
    static void Main(string[] args)
    {
        int fm1, fm2, sm1, sm2;
        //Read Data
        Console.WriteLine("Enter Row size of First Matrix: ");
        fm1 = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter column size of First Matrix: ");
        fm2 = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter Row size of Second Matrix: ");
        sm1 = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter column size of Second Matrix: ");
        sm2 = Convert.ToInt32(Console.ReadLine());

        int[,] firstmatrix = new int[fm1, fm2];
        int[,] secondmatrix = new int[sm1, sm2];
        int[,] productmatrix = new int[fm1, sm2];

        //first matrix
        for(int i = 0; i < fm1; i++)
        {
            for(int j = 0; j < fm2; j++)
            {
                Console.Write($"Enter number at ({i},{j}): ");
                firstmatrix[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }

        for (int i = 0; i < fm1; i++)
        {
            for(int j = 0; j < fm2; j++)
            {
                Console.Write($"{firstmatrix[i, j]} ");
            }
            Console.WriteLine("\n");
        }

        //second matrix
        for (int i = 0; i < sm1; i++)
        {
            for (int j = 0; j < sm2; j++)
            {
                Console.Write($"Enter number at ({i},{j}): ");
                secondmatrix[i, j] = Convert.ToInt32(Console.ReadLine());
            }
        }

        for (int i = 0; i < sm1; i++)
        {
            for(int j = 0; j < sm2; j++)
            {
                Console.Write($"{secondmatrix[i, j]} ");
            }
        }
    }
}
```



```

        Console.WriteLine("\n");
    }

    //product matrix
    if(fm2 != sm1)
    {
        Console.WriteLine("Product of Matrix is not possible");
    }
    else
    {
        for (int i = 0; i < fm1; i++)
        {
            for (int j = 0; j < sm2; j++)
            {
                productmatrix[i, j] = 0;
                for(int k= 0; k < fm2; k++)
                {
                    productmatrix[i, j] += firstmatrix[i, k] *
secondmatrix[k, j];
                }
            }
        }
        Console.WriteLine("Product of the two matrix: ");

        for(int i=0;i<fm1;i++)
        {
            for(int j=0;j<sm2;j++)
            {
                Console.Write($"{productmatrix[i, j]} ");
            }
            Console.WriteLine("\n");
        }
        Console.ReadLine();
    }
}
}
}

```

Output:

```
2
Enter Row size of Second Matrix:
2
Enter column size of Second Matrix:
3
Enter number at (0,0): 11
Enter number at (0,1): 12
Enter number at (1,0): 13
Enter number at (1,1): 14
Enter number at (2,0): 15
Enter number at (2,1): 16
11 12
13 14
15 16
Enter number at (0,0): 17
Enter number at (0,1): 18
Enter number at (0,2): 19
Enter number at (1,0): 20
Enter number at (1,1): 21
Enter number at (1,2): 22
17 18 19
20 21 22
Product of the two matrix:
427 450 473
501 528 555
575 606 637
```

7. What is a jagged array.
What is the benefit of jagged array?

- A Jagged array is a 2D-Array saving different sizes.
- A jagged array is an array whose elements are arrays.
- A jagged array sometimes calls as array of arrays.
- Jagged array saves memory.

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


Code:

```
//Author: Bhanu Prakash Reddy
//WACP for jagged array
internal class Program
{
    static void Main(string[] args)
    {
        char[][] names = new char[3][];

        names[0] = new char[] { 'K', 'O', 'B', 'E' };
        names[1] = new char[] { 'L', 'E', 'B', 'R', 'O', 'N' };
        names[2] = new char[] { 'J', 'O', 'R', 'D', 'O', 'N' };

        //Array elements
        for (int i = 0; i < names.Length; i++)
        {
            Console.WriteLine($"Name at position {i} : ");
            for (int j = 0; j < names[i].Length; j++)
            {
                Console.Write(names[i][j]);
            }
            Console.WriteLine("\n");
        }
        Console.ReadLine();
    }
}
```

Output:

 D:\assignments\Day13Assignment8\Day13Assignment8\bin\Debug\Day13Assignment8.exe

```
Name at position 0 : KOBE
Name at position 1 : LEBRON
Name at position 2 : JORDON
```

9. What is Recursion.

- A function calling itself repeatedly until a specified condition is satisfied.
- This will call function with parameters and receive new parameter after every execution.

10. WACP to illustrate usage of Recursion.

Code:

```
//Authir: Bhanu Prakash Reddy
//WACP for Recursion
internal class Program
{
    class Factorial
    {
        public int Fact(int number)
        {
            if (number == 1)
                return 1;
            else
                return number * Fact(number - 1);
        }
    }


    static void Main(string[] args)
    {
        int input;

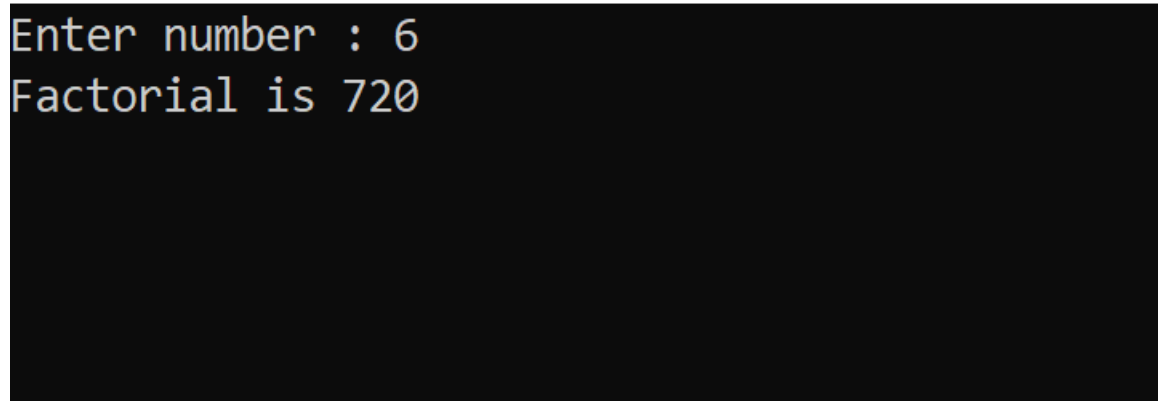
        Console.Write("Enter number : ");
        input = Convert.ToInt32(Console.ReadLine());

        Factorial fact = new Factorial();
        Console.WriteLine($"Factorial is {fact.Fact(input)}");

        Console.ReadLine();
    }
}
```

Output:

 D:\assignments\Day13 Assignment10\Day13 Assignment10\bin\Debug\Day13 Assignment10.exe



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

- Stack is a special type of collection that stores elements in LIFO style => Last In First Out.
- POP – Removes the elements and return the elements.
- PEEK – Not remove the elements and return the elements.

Code:

```
//Authir: Bhanu Prakash Reddy
//WACP for Recursion
internal class Program
{
    class Factorial
    {
        public int Fact(int number)
        {
            if (number == 1)
                return 1;
            else
                return number * Fact(number - 1);
        }
    }


    static void Main(string[] args)
    {
        int input;

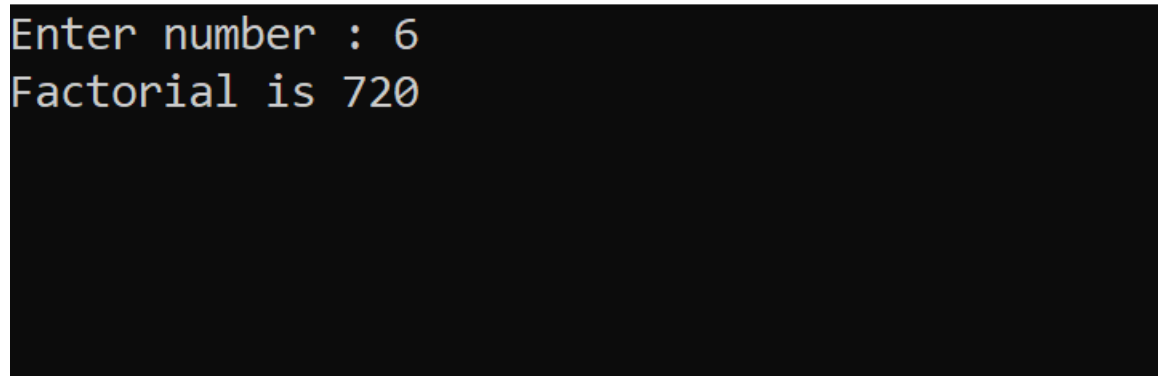
        Console.Write("Enter number : ");
        input = Convert.ToInt32(Console.ReadLine());

        Factorial fact = new Factorial();
        Console.WriteLine($"Factorial is {fact.Fact(input)}");

        Console.ReadLine();
    }
}
```

Output:

 D:\assignments\Day13 Assignment10\Day13 Assignment10\bin\Debug\Day13 Assignment10.exe



```
Enter number : 6
Factorial is 720
```

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

- Queue represents a first-in, first-out collection of objects.
- It is used when you need a first-in, first-out access of items.


Code:

```
//Author: Bhanu Prakash Reddy
//WACP for Stack
internal class Program
{
    static void Main(string[] args)
    {
        Stack<int> data = new Stack<int>();
        data.Push(11);
        data.Push(12);
        data.Push(13);

        for (int i = 0; i <= data.Count; i++)
            Console.WriteLine($"Element POP {data.Pop()}");
        Console.WriteLine($"Element peek {data.Peek()}");

        Console.ReadLine();
    }
}
```

Output:

 D:\assignments\Day13assignment11\Day13assignment11\bin\Debug\Day13assignment11.exe

```
Element POP 13
Element POP 12
Element peek 11
_
```