Day 13 - Assignment

By Manoj Karnatapu - NBHealthCareTechnologies

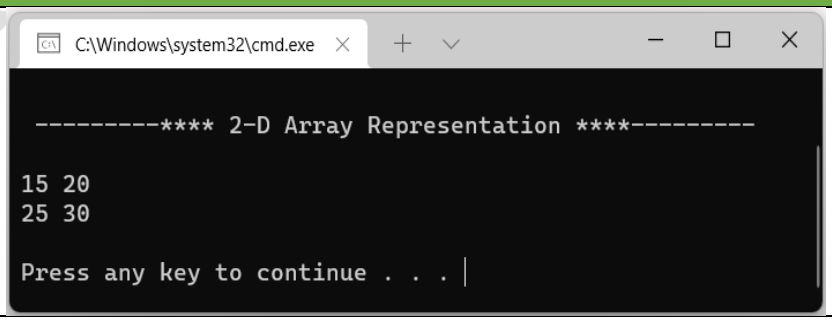
Assignment 1

Declare a 2-D Arrays of size (2x2), initialize using indexes' & print using nested for loop.

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Declaring (2x2) 2-D Array using indexes' & Printing the Value
using nested for Loops.

// For Reference, check Day13Project1 in the same Repository.
namespace Day13Project1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // initialigation & Declaring 2-D Array using Indexes.
            int[,] data = new int[2, 2];
            data[0, 0] = 15;
            data[0, 1] = 20;
            data[1, 0] = 25;
            data[1, 1] = 30;

            // Printing the 2-D Array, Which is Created.
            Console.WriteLine("\n ---------**** 2-D Array Representation ****--
-------\n");
            for(int i = 0; i < 2; i++)
            {
                for(int j = 0; j < 2; j++)
                {
                    Console.Write(data[i, j] + " ");
                }
                Console.Write("\n");
            }

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

Output



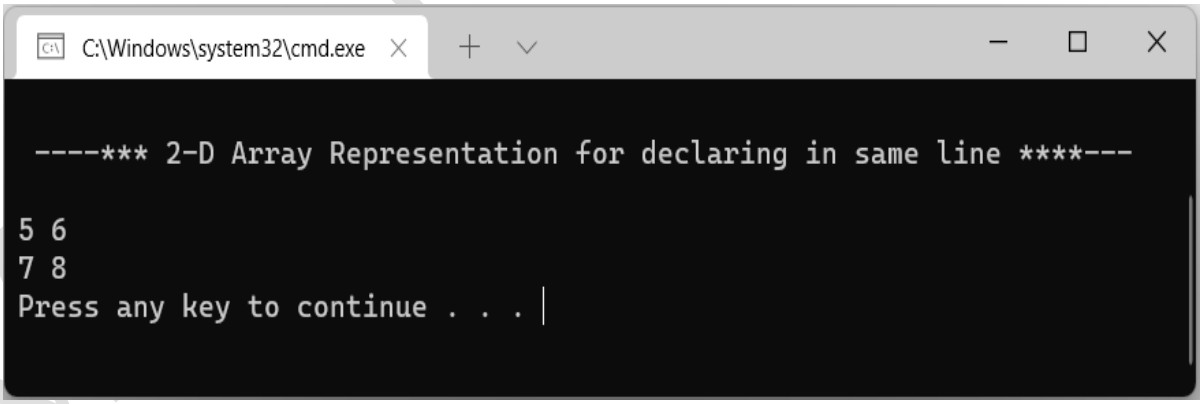
Assignment 2

Declare a 2-D Arrays of size (3x2), & declare with initialization. Print using nested for loop ?

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Declaring (3x2) 2-D Array in the same line while declaring & Printing the
Values using nested for Loops.

// For Reference, check Day13Project2 in the same Repository.
namespace Day13Project2
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // initializing & Declaring 2-D Array
            int[,] data = new int[,] { { 5, 6 }, { 7, 8 }, { 9, 10 } };

            // printing the 2-D Array
            Console.WriteLine("\n ----*** 2-D Array Representation for declaring in same
line ****---\n");
            for (int i=0; i < 2;i++)
            {
                for(int j=0; j < 2;j++)
                {
                    Console.Write(data[i,j] + " ");
                }
                Console.Write("\n");
            }
        }
    }
}
```

Output



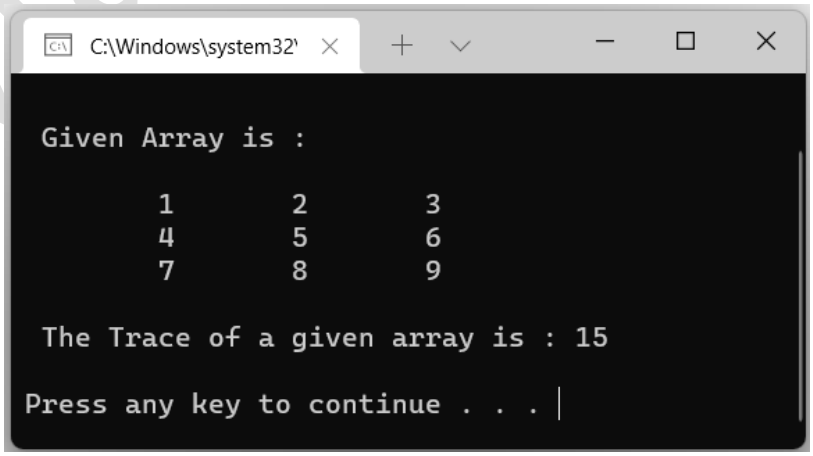
Assignment 3

Declare a 2-D Arrays of size (3x3), & print the Trace of an Array ?

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Declaring (3x3) 2-D Array & Print the Trace of an Array.

// For Reference, check Day13Project3 in the same Repository.
namespace Day13Project3
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data = new int[,] { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };
            int sum = 0;

            Console.WriteLine("\n Given Array is : \n");
            for (int i = 0;i<3;i++)
            {
                for(int j = 0;j<3;j++)
                {
                    Console.Write("\t" + data[i,j] + " ");
                }
                Console.Write("\n");
            }

            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 The Trace of a given array is : {0}",sum);
            Console.ReadLine();
        }
    }
}
```

Output



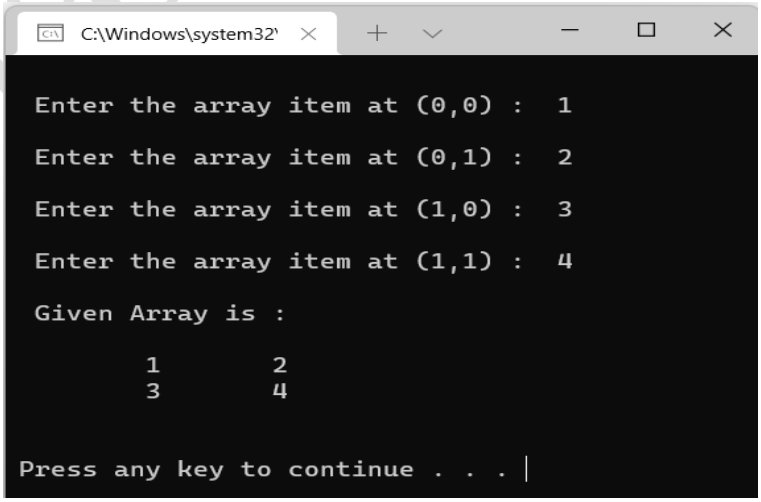
Assignment 4

Declare a 2-D Arrays of size (2x2), read & print the Array values ?

Code

```
using System;
// Author : Manoj.Karnatapu
// Purpose : Declaring (2x2) 2-D Array, Read values from user & Print the
Array.

// For Reference, check Day13Project4 in the same Repository.
namespace Day13Project4
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data = new int[2, 2];

            // Reading Array Items from the user
            for(int i = 0; i < 2; i++)
            {
                for(int j = 0; j < 2; j++)
                {
                    Console.Write($"\n Enter the array item at ({i},{j}) :  ");
                    data[i,j] = Convert.ToInt32(Console.ReadLine());
                }
            }

            // Printing the Array Values into the Console.
            Console.WriteLine("\n Given Array is : \n");
            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.Write("\t" + data[i, j] + " ");
                }
                Console.Write("\n");
            }
            Console.WriteLine("\n");
        }
    }
}
```

Output



Assignment 5

Declare a TWO 2-D Arrays of size (2x2), read & print the sum of two matrices ?

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Declare TWO (2x2) 2-D Arrays, Read values from user & Print the
Sum of Two Matrices.

// For Reference, check Day13Project5 in the same Repository.
namespace Day13Project5
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Program obj = new Program();
            obj.SumOfTwoArrays();
            Console.ReadLine();
        }
        void SumOfTwoArrays()
        {
            Console.Write("Enter Number to Define Rows & Column:- ");
            int arrayLength = Convert.ToInt32(Console.ReadLine());

            int[,] array = new int[arrayLength, arrayLength];
            int[,] arraySecond = new int[arrayLength, arrayLength];
            int[,] arraySum = new int[arrayLength, arrayLength];

            for (int i = 0; i < arrayLength; i++)
            {
                for (int j = 0; j < arrayLength; j++)
                {
                    Console.Write("Array Index [{0}][{1}]:- ", i, j);
                    array[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }

            Console.WriteLine("-------------------------------------");

            Console.WriteLine("This is Your First Array:-");

            for (int i = 0; i < arrayLength; i++)
            {
                for (int j = 0; j < arrayLength; j++)
                {
                    if (j == 0)
                    {
                        Console.Write(array[i, j]);
                    }
                    else
                    {
                        Console.Write(" " + array[i, j]);
                    }

                }
                Console.WriteLine();
            }

            Console.WriteLine("-------------------------------------");

            Console.WriteLine("Now Enter Your Second Array");
```

```
            for (int i = 0; i < arrayLength; i++)
            {
                for (int j = 0; j < arrayLength; j++)
                {
                    Console.Write("Array Index [{0}][{1}]:- ", i, j);
                    arraySecond[i, j] = Convert.ToInt32(Console.ReadLine());
                }
            }

            Console.WriteLine("-------------------------------------");

            Console.WriteLine("This is Your Second Array:-");

            for (int i = 0; i < arrayLength; i++)
            {
                for (int j = 0; j < arrayLength; j++)
                {
                    if (j == 0)
                    {
                        Console.Write(arraySecond[i, j]);
                    }
                    else
                    {
                        Console.Write(" " + arraySecond[i, j]);
                    }

                }
                Console.WriteLine();
            }

            Console.WriteLine("----------------------------------------");
            Console.WriteLine("Do you want to add this arrays:- (Y/N)");

            string userInput = Convert.ToString(Console.ReadLine());

            if (userInput.ToUpper() == "Y")
            {
                for (int i = 0; i < arrayLength; i++)
                {
                    for (int j = 0; j < arrayLength; j++)
                    {
                        arraySum[i, j] = array[i, j] + arraySecond[i, j];
                    }
                }
                Console.WriteLine("-------------------------------------");
                Console.WriteLine("Array is Added Successfully Here is your
Result");
                Console.WriteLine("-------------------------------------");

                for (int i = 0; i < arrayLength; i++)
                {
                    for (int j = 0; j < arrayLength; j++)
                    {
                        if (j == 0)
                        {
                            Console.Write(arraySum[i, j]);
                        }
                        else
                        {
                            Console.Write(" " + arraySum[i, j]);
                        }

                    }
                    Console.WriteLine();
                }
            }
```

```
            else
            {
                Console.WriteLine("Program Terminate Press Enter To
Exit.............");
            }
        }
    }
}
```

Output

```
C:\Windows\system32\cmd.exe

Enter Number to Define Rows & Column:- 2
Array Index [0][0]:- 1
Array Index [0][1]:- 2
Array Index [1][0]:- 3
Array Index [1][1]:- 4
-----------------------------------------------------------------

This is Your First Array:-
1 2
3 4
-----------------------------------------------------------------

Now Enter Your Second Array
Array Index [0][0]:- 5
Array Index [0][1]:- 6
Array Index [1][0]:- 7
Array Index [1][1]:- 8
-----------------------------------------------------------------

This is Your Second Array:-
5 6
7 8
-----------------------------------------------------------------

Do you want to add this arrays:- (Y/N)
Y
-----------------------------------------------------------------

Array is Added Successfully Here is your Result
-----------------------------------------------------------------

6 8
10 12

Press any key to continue . . .
```

Assignment 6

Declare a TWO 2-D Arrays of size (2x2), read & print the product of two matrices ?

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Declare Two 2-D Arrays of Any symmetrical size,Read values form
user & Print Product of Two Matrices.

// For Reference, Check Day13Project6 in the same Repository.
namespace Day13Project6
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int i, j, k, r1, c1, r2, c2, sum = 0;

            int[,] arr1 = new int[50, 50];
            int[,] brr1 = new int[50, 50];
            int[,] crr1 = new int[50, 50];

            Console.Write("\n\n\tMultiplication of two Matrices");
            Console.Write("\n-----------------------------------------------\n");

            Console.Write("\nInput the number of rows and columns of the first
matrix :\n");
            Console.Write("Rows : ");
            r1 = Convert.ToInt32(Console.ReadLine());
            Console.Write("Columns : ");
            c1 = Convert.ToInt32(Console.ReadLine());

            Console.Write("\nInput the number of rows of the second matrix
:\n");
            Console.Write("Rows : ");
            r2 = Convert.ToInt32(Console.ReadLine());
            Console.Write("Columns : ");
            c2 = Convert.ToInt32(Console.ReadLine());

            if (c1 != r2)
            {
                Console.Write("Mutiplication of Matrix is not possible.");
                Console.Write("\nColumn of first matrix and row of second
matrix must be same.");
            }
            else
            {
                Console.Write("Enter the Input elements in the first matrix
:\n");

                for (i = 0; i < r1; i++)
                {
                    for (j = 0; j < c1; j++)
                    {
                        Console.Write($"element - [{i}],[{j}] : ");
                        arr1[i, j] = Convert.ToInt32(Console.ReadLine());
                    }
                }

                Console.Write("\nThe First matrix is :\n");
                for (i = 0; i < r1; i++)
                {
                    Console.Write("\n");
```

```
                    for (j = 0; j < c1; j++)
                        Console.Write("{0}\t", arr1[i, j]);
                }

                Console.Write("\n\nEnter the Input elements in the second
matrix :\n\n");
                for (i = 0; i < r2; i++)
                {
                    for (j = 0; j < c2; j++)
                    {
                        Console.Write("element - [{0}],[{1}] : ", i, j);
                        brr1[i, j] = Convert.ToInt32(Console.ReadLine());
                    }
                }

                Console.Write("\nThe Second matrix is :\n");
                for (i = 0; i < r2; i++)
                {
                    Console.Write("\n");
                    for (j = 0; j < c2; j++)
                        Console.Write("{0}\t", brr1[i, j]);
                }
                Console.Write("\n");

                //multiplication of matrix
                for (i = 0; i < r1; i++)
                    for (j = 0; j < c2; j++)
                        crr1[i, j] = 0;
                for (i = 0; i < r1; i++)    //row of first matrix
                {
                    for (j = 0; j < c2; j++)    //column of second matrix
                    {
                        sum = 0;
                        for (k = 0; k < c1; k++)
                            sum = sum + arr1[i, k] * brr1[k, j];
                        crr1[i, j] = sum;
                    }
                }
                Console.Write("\nThe multiplication of two matrix is : \n");
                for (i = 0; i < r1; i++)
                {
                    Console.Write("\n");
                    for (j = 0; j < c2; j++)
                    {
                        Console.Write("{0}\t", crr1[i, j]);
                    }
                }
            }
            Console.Write("\n\n");
        }
    }
}
```

Output

```
C:\Windows\system32\cmd.exe

        Multiplication of two Matrices
-----------------------------------------

Input the number of rows and columns of the first matrix :
Rows : 2
Columns : 2

Input the number of rows of the second matrix :
Rows : 2
Columns : 2
Enter the Input elements in the first matrix :
element - [0],[0] : 1
element - [0],[1] : 2
element - [1],[0] : 3
element - [1],[1] : 4

The First matrix is :

1       2
3       4

Enter the Input elements in the second matrix :

element - [0],[0] : 5
element - [0],[1] : 6
element - [1],[0] : 7
element - [1],[1] : 8

The Second matrix is :

5       6
7       8

The multiplication of two matrix is :

19      22
43      50

Press any key to continue . . .
```

Assignment 7

What is jagged array & benefits of array ?

Answer

Definition :

A jagged array is an array whose elements are arrays, possibly of different sizes. A jagged array is sometimes called an "array of arrays." Such that member arrays can be of different sizes.

Uses of Jagged arrays :

- It makes things easy where there is a need to store data in a multi-dimensional way using the same variable name.
- It helps in memory management which makes the program to be executed very smoothly and fast as well.

Assignment 8

Write a C# Code to declare a jagged array & print values ?

Code

```
using System;

// Author : Manoj.Karnatapu
// Purpose : Declaring a jagged array & Printing the value.

// For Reference, Check Day13Project7 in the same repository.

namespace Day13Project7
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // Declaring Jagged Array, and initializing.
            char[][] names = new char[4][];
            names[0] = new char[] {'M', 'a', 'n','o', 'j'};
            names[1] = new char[] { 'P', 'a', 'v', 'a', 'n' };
            names[2] = new char[] { 'V', 'i', 'h', 'a', 'r' };
            names[3] = new char[] { 'V', 'i', 'n', 'a', 'y' };

            // Printing The Jagged Array Values.
            Console.WriteLine("\n --------*** Jagged Array Of Names ***--------
\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");
            }
        }
    }
}
```

Output

```
C:\Windows\system32\cmd.ex

--------*** Jagged Array Of Names ***--------

Manoj

Pavan

Vihar

Vinay

Press any key to continue . . .
```

Assignment 9

What is Recursion ?

Answer

The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called as recursive function.

Assignment 10

Write a C# Code to illustrate Usage of Recursion & Describe benefits of Recursion ?

Answer

Code:

```
using System;

namespace RecursionFunction
{
    0 references
    internal class Program
    {
        1 reference
        public static void PrintOutput(int n)
        {
            Console.WriteLine("\nFactorial of {0} is {1}", n, Factorial(n));
        }
        2 references
        public static int Factorial(int input)
        {
            if (input == 0)
                return 1;
            else
                return input * Factorial(input - 1);
        }
        0 references
        static void Main(string[] args)
        {
            //Variable Declaration Section
            int input;
            Console.WriteLine("\n____:::**** Welcome To Find Factorial Using Recursion ****:::____");
            //Reading Inputs Section
            Console.Write("\n\nEnter any Number, To find It's Factorial : ");
            input = Convert.ToInt32(Console.ReadLine());
            //Program Logic Section

            PrintOutput(input);
            Console.WriteLine("\n\n ____:::*** Developer of this Code is Manoj.Karnatapu© ***:::____");
            Console.ReadLine();
        }
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe

____:::**** Welcome To Find Factorial Using Recursion ****:::____


Enter any Number, To find It's Factorial : 5

Factorial of 5 is 120


 ____:::*** Developer of this Code is Manoj.Karnatapuc ***:::____

Press any key to continue . . .
```

Benefits Of Recursion:

- Recursion can reduce time complexity.
- Recursion reduces unnecessary calling functions.
- Through Recursion one can solve problems in easy way while its iterative solution is very big and complex.
- Extremely useful when applying the same solution.
- Recursion adds clarity and reduces the time needed to write and debug code.
- Recursion is better at tree traversal.

Assignment 11

Write a C# Code to illustrate Usage of Stack & Describe Stack ?

Answer

Code:

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

// Author : Manoj.Karnatapu
// Purpose : Stack Program of implementation using C# Language.
namespace StackQueue
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Stack<int> data = new Stack<int>();
            data.Push(19);
            data.Push(25);
            data.Push(43);
            data.Push(68);
            Console.WriteLine($"The Stack size before removing the last item is : {data.Count}");
            Console.WriteLine($"The Stack item to be removed is : {data.Peek()}");
            Console.WriteLine($"In Stack, the item removed is : {data.Pop()}");
            Console.WriteLine($"The Stack size after removing the last item is : {data.Count}");
            Console.WriteLine($"The Stack next item to be removed is: {data.Peek()}");
            Console.ReadLine();
        }
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
The Stack size before removing the last item is : 4
The Stack item to be removed is : 68
In Stack, the item removed is : 68
The Stack size after removing the last item is : 3
The Stack next item to be removed is: 43

Press any key to continue . . .
```

What is Stack :

- Stack is a Special type of collection that stores elements in LIFO style (Last In First Out).
- Stack is useful to store temporary data in LIFO style, and you might want to delete an element after retrieving its value.

Benefits of Stack :

- Helps you to manage the data in a Last In First Out(LIFO) method which is not possible with Linked list and array.
- It allows you to control how memory is allocated and deallocated.

- Not easily corrupted.
- Stack<T> can contain elements of the specified type. It provides compile-time type checking and doesn't perform boxing-unboxing because it is generic.

Assignment 12

Write a C# Code to illustrate Usage of Queue & Describe Queue ?

Answer

Code:

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

// Author : Manoj.Karnatapu
// Purpose : Queue Program of implementation using C# Language.
namespace StackQueue
{
    0 references
    internal class Program
    {
        0 references
        static void Main(string[] args)
        {
            Queue<int> data = new Queue<int>();
            data.Enqueue(19);
            data.Enqueue(25);
            data.Enqueue(43);
            data.Enqueue(68);
            Console.WriteLine($"The Queue size before removing the first item is : {data.Count}");
            Console.WriteLine($"The Queue item to be removed is : {data.Peek()}");
            Console.WriteLine($"In Queue, the item removed is : {data.Dequeue()}");
            Console.WriteLine($"The Queue size after removing the first item is : {data.Count}");
            Console.WriteLine($"The Queue next item to be removed is: {data.Peek()}");
            Console.ReadLine();
        }
    }
}
```

Output:

```
C:\Windows\system32\cmd.exe
The Queue size before removing the first item is : 4
The Queue item to be removed is : 19
In Queue, the item removed is : 19
The Queue size after removing the first item is : 3
The Queue next item to be removed is: 25

Press any key to continue . . .
```

What is Queue :

- Queue is a Special type of collection that stores elements in FIFO style (First In First Out).
- It contains the elements in the order they were added.

Benefits of Queue :

- We use when you need to perform actions on a set of objects in a sequence.
- It supports multiple readers simultaneously.
- Multiple data can be handled, and they are fast and flexibility.

~~~~~~~~ The End ~~~~~~~~
~~~~~~~~