DAY 13 ASSIGNMENT

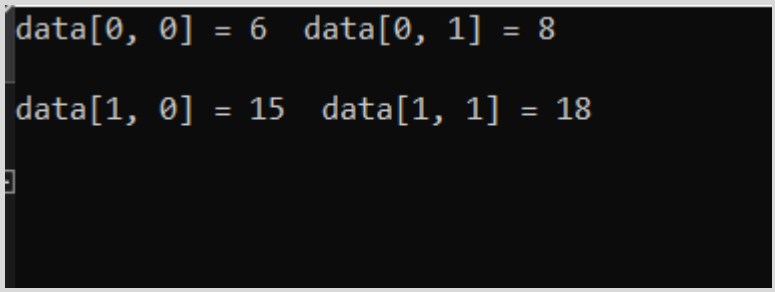
-- BY SARATH KASIMSETTY

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;
//sarath kasimsetty
//Declare a 2 dimentional array of size (2,2) and
//initialize using indexes and print the values using nested for loop

namespace Day13Project1
{
    /// <summary>
    /// 2D array
    /// </summary>
    internal class Program
    {
        static void Main(string[] args)
        {
            /// Size is 2 rows and 2 colums
            int[,] data = new int[2, 2];

            data[0, 0] = 6;
            data[0,1] = 8;
            data[1, 0] = 15;
            data[1, 1] = 18;

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

OUTPUT:

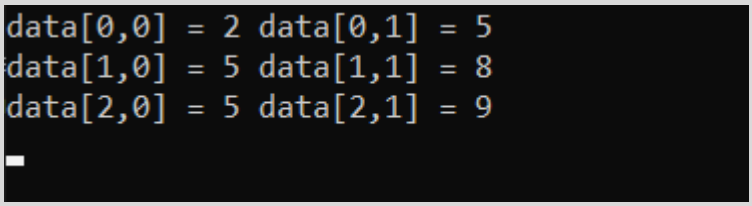


2) Declare a 2-D array of size (3,2) and initialize in the same line while declaring and print the values using nested for loop

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//sarath kasimsetty
//Declare a 2-D array of size (3,2) and
//initialize in the same line while declaring and print the valuesusing nested for
loop

namespace Day13Project2
{
    //2D Array
    internal class Program
    {
        static void Main(string[] args)
        {
            ///size of rows is 3 and colums is 2
            int[,] data = new int[3, 2] { { 2, 5 },{ 5, 8 },{ 5, 9 } };

            for(int i =0;i<3;i++)// i is rows
            {
                for (int j=0;j<2;j++) // j is cols
                {
                    //print the values step by step using loop
                    Console.Write("data[{0},{1}] = {2}",i,j, data[i, j] + " ");
                }
                Console.Write("\n");
            }
            Console.ReadLine();
        }
    }
}
```

OUTPUT:

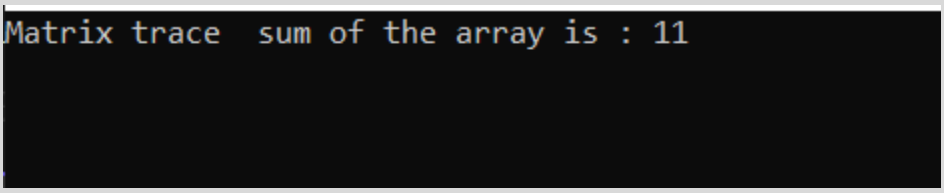


3) Declare a 2-D array of size (3,3) and print trace of the array

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//sarath kasimsetty
//Declare a 2-D array of size (3,3) and print trace of the array
namespace Day13Project3
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[,] data = new int[3, 3] { { 1, 3, 7 }, { 2, 3, 7 }, { 4, 2, 7
} };
            int sum = 0;

            for(int i =0;i<3;i++)
            {
                for(int j=0;j<3;j++)
                {
                    if(i==j) //if same size row and col index is match ex:
[0,0],[1,1],[5,5]
                    {
                        sum = sum + data[i, j];
                    };
                }
            }
            Console.WriteLine("Matrix trace  sum of the array is : {0}",sum);
            Console.ReadLine();
        }
    }
}
```

OUTPUT:



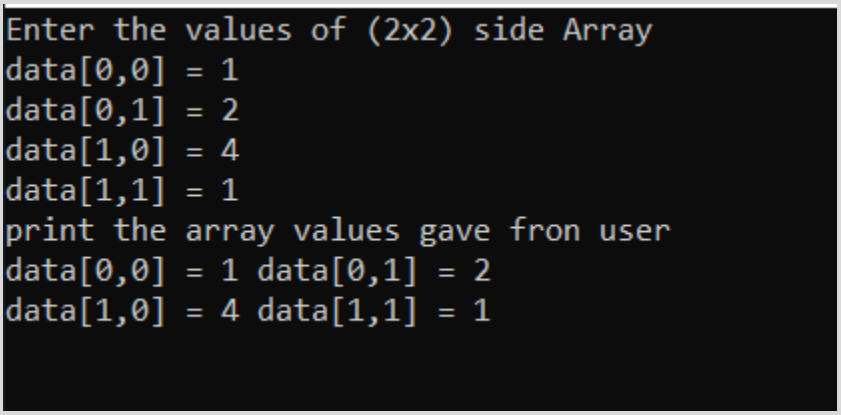
4) Declare a 2-D array of size (2,2) and read values from user and print the array values.

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//sarath kasimsetty
//Declare a 2-D array of size (2,2) and read values from user
//and print the array values.
namespace Day13Project4
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // 2D array of size 2 rows and 2 coloum
            int[,] data = new int[2, 2];

            Console.WriteLine("Enter the values of (2x2) side Array");
            for (int i=0;i<2;i++) // i is row
            {
                for (int j = 0;j<2;j++) // j is coloum
                {
                    Console.Write("data[{0},{1}] = ", i, j);
                    data[i, j] = Convert.ToInt32(Console.ReadLine());
                    //read the value from user
                }
            }

            Console.WriteLine("print the array values gave fron user");
            for (int i = 0; i < 2; i++)
            {
                for(int j=0;j<2;j++)
                {
                    Console.Write("data[{0},{1}] = {2}",i,j,data[i, j] + " ");
                }
                Console.Write("\n");
            }
            Console.ReadLine();
```

```
        }
    }
}
```

OUTPUT:



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;
//sarathkasimsetty
//Declare TWO 2-D arrays of size (2,2) and read values from user
//and print the sum of the two matrices.
namespace Day13Project5
{
    internal class Program
    {
        static void Main(string[] args)
        {
            ///sum of two matrices(size[2,2]) from read value from user.
            int[,] array1 = new int[2, 2];
            int[,] array2 = new int[2, 2];
            int[,] sum = new int[2, 2];

            Console.Write("Input elements in the first matrix :\n");
            for (int i=0;i<2;i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.Write("index[{0},{1}] =", i, j);
```

```
                array1[i,j]=Convert.ToInt32(Console.ReadLine());
                //values are read from user
            }
        }

        for ( int i =0;i<2;i++)
        {
            for(int j =0;j<2;j++)
            {
                //print first matrix from user given values
                Console.Write(array1[i, j]+" ");
            }
            Console.WriteLine("\n");
        }

        Console.Write("Input elements in the second matrix :\n");
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 2; j++)
            {
                Console.Write("index[{0},{1}] =", i, j);
                array2[i, j] = Convert.ToInt32(Console.ReadLine());
                //values are read from user
            }
        }

        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 2; j++)
            {
                Console.Write( array2[i, j]+" ");
                ////print second matrix from user given values
            }
            Console.WriteLine("\n");
        }

        for(int i = 0;i<2;i++)
        {
            for(int j=0;j<2;j++)
            {
                //sum of the first matrix and second matrix
                sum[i, j] = array1[i, j] + array2[i, j];
            }
        }

        Console.Write("Sum of the two matrix :\n");
        for (int i = 0; i < 2; i++)
        {
            for (int j = 0; j < 2; j++)
            {
                Console.Write(sum[i, j] + " ");
            }
            Console.WriteLine("\n");
        }
        Console.ReadLine();
```

```
        }
    }
}
```

OUTPUT:

```
Input elements in the first matrix :
index[0,0] =1
index[0,1] =2
index[1,0] =3
index[1,1] =4
1 2

3 4

Input elements in the second matrix :
index[0,0] =4
index[0,1] =3
index[1,0] =2
index[1,1] =1
4 3

2 1

Sum of the two matrix :
5 5

5 5
```

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;

namespace Day13Project6
{
    internal class Program
    {
        static void Main(string[] args)
        {
            ///product of two matrices(size[2,2]) from read value from user.
            int[,] array1 = new int[2, 2];
            int[,] array2 = new int[2, 2];
            int[,] product = new int[2, 2];

            Console.Write("Input elements in the first matrix :\n");
            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.Write("index[{0},{1}] =", i, j);
                    array1[i, j] = Convert.ToInt32(Console.ReadLine());
                    //values are read from user
                }
            }

            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    //print first matrix from user given values
                    Console.Write(array1[i, j] + " ");
                }
                Console.WriteLine("\n");
            }

            Console.Write("Input elements in the second matrix :\n");
            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.Write("index[{0},{1}] =", i, j);
                    array2[i, j] = Convert.ToInt32(Console.ReadLine());
                    //values are read from user
                }
            }

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

```
                    Console.Write(array2[i, j] + " ");
                    ////print second matrix from user given values
                }
                Console.WriteLine("\n");
            }

            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    //product of the first matrix and second matrix
                    product[i, j] = array1[i, j] * array2[i, j];
                }
            }

            Console.Write("product of the two matrix :\n");
            for (int i = 0; i < 2; i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    Console.Write(product[i, j] + " ");
                }
                Console.WriteLine("\n");
            }
            Console.ReadLine();
        }
    }
}
```

OUTPUT:

```
Input elements in the first matrix :
index[0,0] =1
index[0,1] =2
index[1,0] =3
index[1,1] =4
1 2

3 4

Input elements in the second matrix :
index[0,0] =4
index[0,1] =2
index[1,0] =1
index[1,1] =3
4 2

1 3

product of the two matrix :
4 4

3 12
```

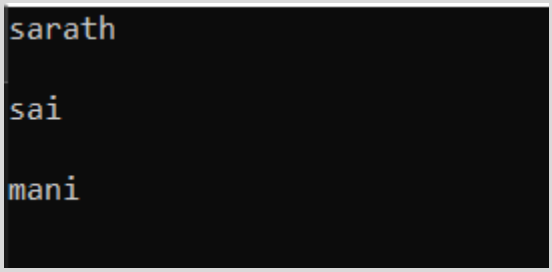
8) WACP to declare a jagged array and print values

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//sarath kasimsetty
//WACP to declare a jagged array and print values

namespace Day13Project8
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // Declaring Jagged Array, and initializing.
            char[][] names = new char[3][];
```

```
            names[0] = new char[] { 's', 'a', 'r', 'a', 't', 'h' };
            names[1] = new char[] { 's', 'a', 'i' };
            names[2] = new char[] { 'm', 'a', 'n', 'i' };

            //printing jagged array values
            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:



9) What is Recursion.

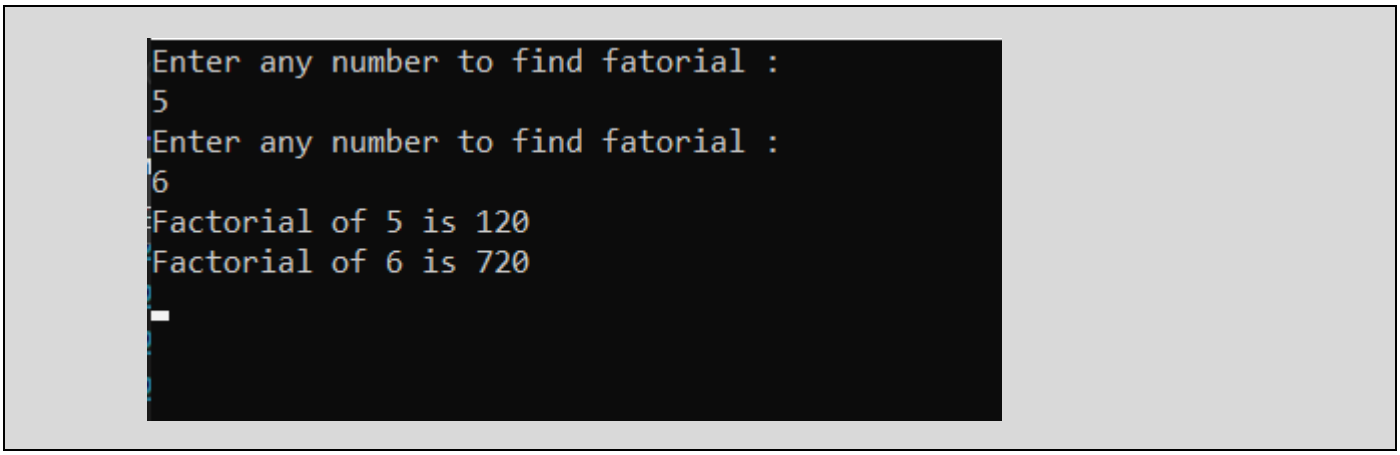
- The process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called as recursive function.

10) WACP to illustrate usage of Recursion.

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//sarath kasimsetty
//WACP to illustrate usage of Recursion.

namespace Day13Project10
{
    internal class Program
    {
        static void PrintOutput(int a)
        {
            Console.WriteLine("Factorial of {0} is {1} ", a, Factorial(a));
        }
        static int Factorial(int a) //Function parameter
        {
            if (a == 0)
                return 1;
            else
                return a * Factorial(a - 1);
        }
        static void Main(string[] args)
        {
            int num1,num2;
            //read values from user
            Console.WriteLine("Enter any number to find fatorial : ");
            num1 = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Enter any number to find fatorial : ");
            num2 = Convert.ToInt32(Console.ReadLine());

            //print output from funtion return
            PrintOutput(num1);
            PrintOutput(num2);

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

OUTPUT:



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

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//srath kasimsetty
//WACP to illustrate usage of Stack<>
//Write couple of points about Stack

namespace Day13Project11
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Stack<int> data = new Stack<int>();
            data.Push(15);
            data.Push(21);
            data.Push(10);

            Console.WriteLine(data.Count());
            Console.WriteLine(data.Peek());
            Console.WriteLine(data.Count());
            Console.WriteLine(data.Pop());
            Console.WriteLine(data.Count());

            Console.ReadLine();
        }
    }
```

```
}

// 1) Helps you to manage the data in a Last In First Out method which is not
//    possible with Linked list and array so , by using Stack generics.
// 2) Stack is useful to store temporary data in (Last In First Out) , and you
might want to
//    delete an element after retrieving its value.
// 3) pop is remove the element and count vlaue also remove.
// 4) peek is only remove temporary valve of element.
```

OUTPUT:

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

CODE:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//sarath kasimsetty
//WACP to illustrate usage of Queue<>
//Write couple of points about Stack

namespace Day13Project12
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Queue<int> data = new Queue<int>();

            data.Enqueue(19);
            data.Enqueue(15);
            data.Enqueue(14);
            data.Enqueue(10);
            Console.WriteLine(data.Count());
            Console.WriteLine(data.Peek());
```

```
            Console.WriteLine(data.Count());
            Console.WriteLine(data.Dequeue());
            Console.WriteLine(data.Count());

            Console.ReadLine();
        }
    }
}
// 1) Queue is a Special type of collection that stores elements in (First In
//    First Out).
// 2) peek is the temporary element and return that the element
//    will be first.
// 3) DeQueue is the remove the element and also count is remove.
```

OUTPUT: