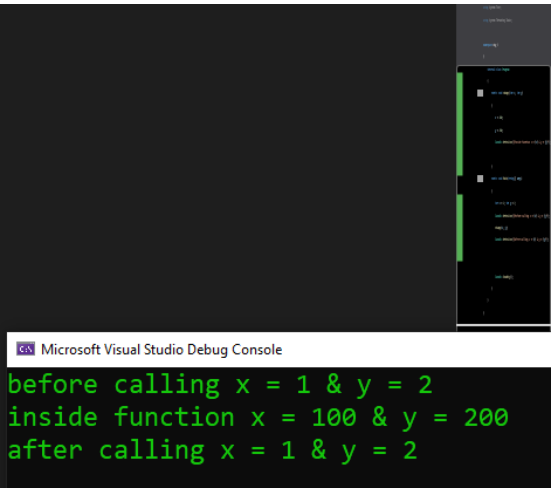
DAY 3 C#

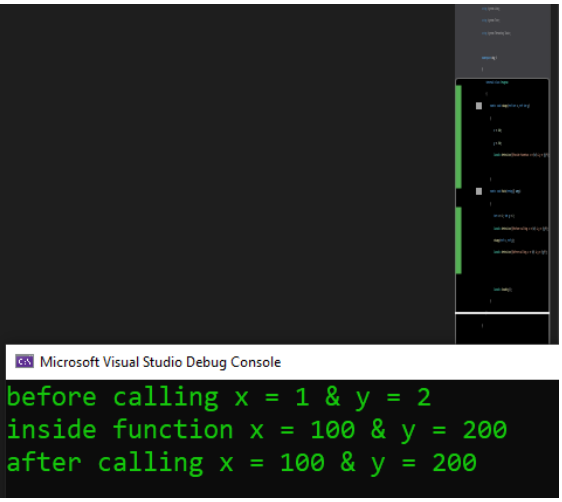
```
1 reference
static void change(int x, int y)
{
    x = 100;
    y = 200;
    Console.WriteLine($"inside function x = {x} & y = {y}");

}
0 references
static void Main(string[] args)
{
    int x = 1; int y = 2;
    Console.WriteLine($"before calling x = {x} & y = {y}");
    change(x, y);
    Console.WriteLine($"after calling x = {x} & y = {y}");
```



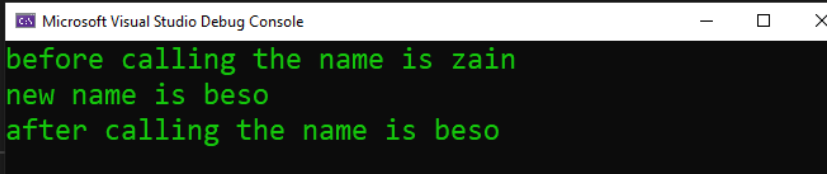

```
1 reference
static void change(ref int x, ref int y)
{
    x = 100;
    y = 200;
    Console.WriteLine($"inside function x = {x} & y = {y}");

}
0 references
static void Main(string[] args)
{
    int x = 1; int y = 2;
    Console.WriteLine($"before calling x = {x} & y = {y}");
    change(ref x, ref y);
    Console.WriteLine($"after calling x = {x} & y = {y}");
```




```
3 references
class names
{
    public string name;
}
0 references
internal class Program
{
    1 reference
    static void change(names FirstName)
    {
        FirstName.name = "beso";
        Console.WriteLine($"new name is {FirstName.name}");
    }
    0 references
    static void Main(string[] args)
    {
        names FirstName = new names();
        FirstName.name = "zain";
        Console.WriteLine($"before calling the name is {FirstName.name}");
        change( FirstName );
        Console.WriteLine($"after calling the name is {FirstName.name}");

        Console.ReadKey();
    }
}
```



```
3 references
class names
{
    public string name;
}
0 references
internal class Program
{
    1 reference
    static void change(ref names FirstName)
    {
        FirstName.name = "beso";
        Console.WriteLine($"new name is {FirstName.name}");
    }
    0 references
    static void Main(string[] args)
    {
        names FirstName = new names();
        FirstName.name = "zain";
        Console.WriteLine($"before calling the name is {FirstName.name}");
        change(ref FirstName );
        Console.WriteLine($"after calling the name is {FirstName.name}");

        Console.ReadKey();
    }
}
```

```
Microsoft Visual Studio Debug Console
before calling the name is zain
new name is beso
after calling the name is beso
```

```
2 references
static bool IsPrime(int num)
{
    if (num <= 1) return false;
    for (int i = 2; i < num; i++)
    {
        if (num % i == 0) return false;
    }
    return true;
}
0 references
static void Main(string[] args)
{
    Console.WriteLine("enter number");
    int x = Convert.ToInt32(Console.ReadLine());
    IsPrime(x);
    if(IsPrime(x) == true)
    {
        Console.WriteLine($"{x} is a prime number");
    }else
    {
        Console.WriteLine($"{x} is not a prime number");
    }
```

```
Microsoft Visual Studio Debug Console
enter number
13
13 is a prime number
```

```
1 reference
static void MinMaxArray(int[] array, ref int max, ref int min)
{
    min = 0;
    max = 0;
    for (int i = 0; i < array.Length; i++) {
        if (array[i] > max)
        {
            max = array[i];
        }
        if (array[i] < min) {
            min = array[i];
        }
    }

}
0 references
static void Main(string[] args)
{
    int[] array = { 2, 5, -9, 6, 41 };
    int max = 0;
    int min = 0;
    MinMaxArray(array, ref max, ref min);
    Console.WriteLine($"{max}, {min}");
```

```
Microsoft Visual Studio Debug Console
41, -9
```

```
9 references
enum TrafficLight
{
    Red, Yellow, Green
}
2 references
class TrafficColor
{
    1 reference
    public string changeWithen30Seconds(ref TrafficLight light)
    {
        string output = "";
        switch (light)
        {
            case TrafficLight.Red:
                output = "WATCH OUT! IT IS RED";
                light = TrafficLight.Yellow;
                break;
            case TrafficLight.Yellow:
                output = "WAIT! IT IS GONNA BE GREEN";
                light = TrafficLight.Green;
                break;
            case TrafficLight.Green:
                output = "IT IS OK! YOU CAN GO";
                light = TrafficLight.Red;
                break;
        }
        return output;
    }
}
0 references
internal class Program
{
    0 references
    static void Main(string[] args)
    {
        TrafficColor trafficColor = new TrafficColor();
        TrafficLight light = TrafficLight.Red;

        for (int i = 0; i < 3; i++) {
            Console.WriteLine($"current light : {light}");
            Console.WriteLine(trafficColor.changeWithen30Seconds(ref light));
        }
```

```
Microsoft Visual Studio Debug Console
current light : Red
WATCH OUT! IT IS RED
current light : Yellow
WAIT! IT IS GONNA BE GREEN
current light : Green
IT IS OK! YOU CAN GO

G:\fifth\courses\.NET\c#\c#\first with br
Code1\day 3\bin\Debug\day 3.exe (process
ited with code 0 (0x0).
To automatically close the console when c
stops, enable Tools->Options->Debugging->
ally close the console when debugging sto
Press any key to close this window . . .
```

```
10 references
enum OrderStatus
{
    Pending, Processing, Shipped, Delivered
}
2 references
class order
{
    1 reference
    public string change(ref OrderStatus status)
    {
        string result = "";
        switch(status)
        {
            case OrderStatus.Pending:
                result = "the order is pending";
                status = OrderStatus.Processing;
                break;

            case OrderStatus.Processing:
                result = "the order is Processing";
                status = OrderStatus.Shipped;
                break;

            case OrderStatus.Shipped:
                result = "the order has been shipped";
                status = OrderStatus.Delivered;
                break;

            case OrderStatus.Delivered:
                result = "the order is Delivered";
                break;
        }
        return result;
    }
}
0 references
internal class Program
{
    0 references
    static void Main(string[] args)
    {
        order newOrder = new order();
        OrderStatus status = OrderStatus.Pending;

        for (int i = 0; i < 4; i++) {
            Console.WriteLine(newOrder.change(ref status));
        }

        Console.ReadKey();
```

```
Microsoft Visual Studio Debug Console
the order is pending
the order is Processing
the order has been shipped
the order is Delivered
```