```
void SetValues(ref int a, out int b)
{
    a += 10;
    b = 20;
}

int x = 5;
int y;

SetValues(ref x, out y);
Console.WriteLine($"{x}, {y}");
```

```
using System;
public abstract class Shape
{
    public abstract double CalculateArea();

    public void PrintDetails()
    {
        Console.WriteLine("This is a shape.");
    }
}

public class Rectangle : Shape
{
    public double Width { get; set; }
    public double Height { get; set; }
    public override double CalculateArea()
    {
        return Width * Height;
    }
}

public interface IAnimal
{
    void MakeSound();
}
public class Dog : IAnimal
{
    public void MakeSound()
    {
        Console.WriteLine("Woof!");
    }
}

public class Cat : IAnimal
{
    public void MakeSound()
    {
        Console.WriteLine("Meow!");
    }
}
public class Program
{
    static void Main(string[] args)
    {
        Shape rectangle = new Rectangle();
        rectangle.PrintDetails();
        ((Rectangle)rectangle).Width = 5;
        ((Rectangle)rectangle).Height = 3;
        Console.WriteLine("Area of rectangle: " + rectangle.CalculateArea());

        IAnimal dog = new Dog();
        dog.MakeSound();
        IAnimal cat = new Cat();
        cat.MakeSound();
    }
}
```

```
int[] arr = {1,1,2,2,3,3,4,5,5};
output - 12345

int added = arr[0];
string r = added.ToString();
List<int> result = new List<int>(added);
for(int i = 1; i < arr.Length; i++){
        if(added != arr[i] ){
         r+=arr[i].toString();
        }
added = arr[i];

}
```

```
public class CustomExceptionHandler(RequestDelegate reqDel, IRestApiService iRest)
: IMiddleWare
{

        private readonly  RequestDelegate _reqDel = reqDel;
        public override Task async InvokeAsync(HttpContext httpContext){
                try{
        //use any logic here
//tran

}
        catch(Exception ex){}

        }
}
```