```
class A
{
    public A()
    {
        Console.WriteLine("A's constructor");
    }
public A(string name)
    {
        Console.WriteLine("A's constructor with a parameter");
    }
}
class B : A
{
    public B()
    {
        Console.WriteLine("B's constructor");
    }
public B(string name)
    {
        Console.WriteLine("B's constructor with a parameter");
    }
}

//Called from main
var b = new B("XYZ");
//A's constructor
//B's constructor with a parameter


public static class ExtendTestClass{
        public static string ToLocalPath(this TestClass classVal){

                //do smome logic for TestClass
                string result = "";
                return result;
        }

}
```

Native or Bilingual ProficiencyGiven an integer unorder array of length n. Rearrange the array item in such a way that arr[i] = i and if i is not present then put -1 on that place. Where i is from 0 to n-1.

Example:
Input: arr[] = {3, -1, 0, 5, 2, 14, 7, -1, 5, 4, 10 }
Output: arr[] = {0, -1, 2, 3, 4, 5 , -1, 7, -1, -1, 10 }

```
with cteDupli as (
        select *, row_number() over (partition by name order by (select null)) as
id from table1
)
delete from cteDupli where id > 1


var customers = new List<Customer>

{

    new Customer { Id = 1, Name = "Alice" },

    new Customer { Id = 2, Name = "Bob" },

    new Customer { Id = 3, Name = "Charlie" }

};

var orders = new List<Order>

{

    new Order { Id = 101, CustomerId = 1, Amount = 250 },

    new Order { Id = 102, CustomerId = 2, Amount = 150},

    new Order { Id = 103, CustomerId = 1, Amount = 75}

};
Write a LINQ query to join the two collections and return the CustomerName with
each corresponding OrderAmount


var result = (from c in customers join o in orders on c.Id equals o.CustomerId
select new { c.Name, o.Amount }).ToList();
```