```
using System;
namespace krd
{//practical , without void main
    class Record
    {
       protected string desc;
       private int id;//new object create automatically id=1 ,object 2 id=2

       static int count;
       public Record(string v)//one parameter constructor
       {
           count++;
           id = count;
           desc = v;
       }//
       public virtual void print()
       {
           Console.WriteLine("desk="+desc);
           Console.WriteLine("id="+id);
       }//
    }//end of class Record
    //Rec extend from Record
    class Rec:Record//name class not a problem
    {
        int[] arr;
        public Rec(string v):base(v)
        {
            Console.Write("enter size array : ");
            int a = int.Parse(Console.ReadLine());

            arr=new int[a];

            Console.WriteLine("enter element array !! ");
            for (int i = 0; i < a; i++)
            {
                arr[i] = int.Parse(Console.ReadLine());
            }
        }//constructor
        public override void  print()
        {
            Console.WriteLine("desk="+desc);
            for (int i = 0; i < arr.Length; i++)
            {
                Console.Write(arr[i]+"  ");
            }
            Console.WriteLine();
        }
```

```
    //[1,2,3,2] -> [1,2,3,0] jmara dobara boy bka 0 mrjake j ya daspeke na
    //[1,1,1,2] ->[1,0,0,2]
    //or [1,2,3,4,3] -> [1,2,3,4,0]
    public static Rec operator ~(Rec ob)
    {
        for (int i = 0; i < ob.arr.Length; i++)
        {
            int z = 0;

            for (int j = 0; j < ob.arr.Length; j++)
            {
                if (ob.arr[i] == ob.arr[j])
                {
                    z++;
                    if (z > 1)
                    {
                        ob.arr[j] = 0;
                    }
                }
            }
        }//loop
      return  ob;
    }//end ~
    //2 do aray damege ka wak
    //a=[1,2] , b=[3,4] , c=[1,2,3,4] or c=[3,4,1,2]
    //[1,2] +[4,5,6,7] -> [1,2,4,5,6,7] or [4,5,6,7,1,2]
    public static Rec operator +(Rec ob1 , Rec ob2)
    {
        int size=(ob1.arr.Length)+(ob2.arr.Length);
        int []damage=new int[size];
        int index = 0;
        for (int i = 0; i < size; i++)
        {
            if (i<ob1.arr.Length)
            {
                damage[i] = ob1.arr[i];
            }
            else
            {
                damage[i] = ob2.arr[index];
                index++;
            }
        }//
        ob1.arr = damage;
        return ob1;
    }
}//end class rec
```

```
    class text:Record
    {
        string[] line;//not in my reminder
        public text(string v):base(v)
        {
            Console.Write("enter size array : ");
            int a = int.Parse(Console.ReadLine());

            line= new string[a];

            Console.WriteLine("enter element array !! ");
            for (int i = 0; i < a; i++)
            {
                line[i] = Console.ReadLine();
            }
        }//
        public override void print()
        {
            Console.WriteLine("desk=" + desc);
            for (int i = 0; i < line.Length; i++)
            {
                Console.Write(line[i] + "  ");
            }
            Console.WriteLine();
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            //Rec a = new Rec("krd music");
            //Rec b = new Rec("");
            //a = a + b;
            //a.print();
            //without void main
        }
    }
}
```