Int [] numbers = new int[10]; нов празен масив с десет нули, дължина десет

--

String values = Console.ReadLine();

String [] items = values.Split(‘ ‘);

Int [] arr = new int[items.Length];

For (int I = 0; I < items.Length; i++)

{

Arr[i] = int.Parse(items[i]);

Round away from zero – бягаме от нулата, в посока противоположна на 0

Foreach loop

String.Join

For (int = 0; I < arr.Length; i++)

{

Var element = arr[i]

Cw(element)

}

Съшото като foreach (var element in arr)

{ cw(element)}

.ToArray() – казва че пълним в масив

Count при листа е като Length

Листовете НЯМАТ твърдо дефинирана дължина

Contains – проверява дали даден елемент е вътре в листа

List<int> ---- Count 0

List<int> list = new List<int>();

list.Add(5); - просто добавя числото 5 в списъка – не прави нова променлива, просто добавя в списъка

list.Remove(-7); - премахва елемент ПЪРВИЯ ПЪТ КОГАТО ГО НАМЕРИ (ако се повтаря няколко пъти)

list.RemoveAt(0) – премахва по индекс

list.Insert(2, 5); - вкарва числото 5 на втора позиция

Масивите са много по-бързи като работа! Но пък не могат да се оразмеряват за разлика от листовете.

Сортиране може да се случи само ако елементите могат да бъдат сравнени еднозначно.

Names.Sort();

Добавяне в нов масив int[] array = { 1, 2, 3, 4};

Добавяне в лист -

Var names = new List<string>()

{ “asdasd”, “dsfsgdgs”};

Names.Sort((a, b) => b.CompareTo(a)); - компаратор, казва как да сортираме елементите. В случай сравнява от голямо към малко

Може директно string[] input = Console.ReadLine().Split();

Проверяване за корен квадратен – sqrt = (int)sqrt