

Ordenar Por Cercanía

Dados 3 números enteros **a**, **b** y **p**, se dice que **a** es más cercano a **p** que **b** si y solo si se cumple:

$|a-p| < |b-p|$ donde $|x|$ representa la función módulo.

De cumplirse $|a-p| = |b-p|$ se dice que ambos números, **a** y **b** están a la misma distancia de **p**.

Usted deberá implementar un método que ordene los elementos de un *array* de enteros por orden de cercanía a un entero *p* dado, al que llamaremos pivote. En caso de que existan varios elementos en el *array* a la misma distancia del elemento pivote, estos se deberán ordenar de menor a mayor.

Veamos algunos ejemplos:

Ejemplo 1:

Para el *array* {5, 3, 7, 10} y el pivote $p=7$,

los elementos del *array* quedarían en el siguiente orden: {7, 5, 10, 3}.

pues: $|7-7|=0$; $|5-7|=2$; $|10-7|=3$; $|3-7|=4$;

Ejemplo 2:

Para el *array* {4, 2, 11, 8} y el pivote $p=7$

los elementos del *array* quedarían en el siguiente orden: {8, 4, 11, 2}*.

(*) Note que el elemento pivote puede o no pertenecer al *array*.

Ejemplo 3:

Para el *array* {2, 10, 8, 4} y el pivote $p=7$

los elementos del *array* quedarían en el siguiente orden: {8, 4, 10, 2}**.

(**) Note que los elementos 4 y 10 están a la misma distancia del pivote, sin embargo el orden {8, 10, 4, 2} no es correcto ya que $4 < 10$.

Implemente una biblioteca de clases de nombre **ExamenOrdenacion** (al compilarse produce un ensamblado **ExamenOrdenacion.dll**). Su método deberá tener la siguiente signatura:

```
namespace ExamenOrdenacion
{
    public class Utiles
    {
        public static void OrdenarPorCercania(int[]elementos, int pivote)
        {
            ...
        }
    }
}
```