

Relación de Ejercicios de Arrays (4)

En esta relación se requiere modificar el tamaño de un array sobre la marcha. Esto se hace con la función `Array.Resize(ref <array>, <tamaño>)`.

31. Escribe una función “ConcatenaArraysPro” a la que le pasas dos arrays de enteros y te devuelve un array cuyo tamaño es la suma del tamaño de ambos y que contiene todos los elementos del primero y a continuación los del segundo. Esta función devolverá un array de enteros (`int[]`) el cuál se creará dentro de la propia función.

Ej. de uso:

```
int[] a = {1,2,3};
int[] b = {4,5,6,7,2};
int[] c;
c = ConcatenaArraysPro(a, b);
```

32. Escribe una función “CopiaArrayPro” a la que le pasas un array y te devuelve un array del mismo tamaño y con los mismos datos.

Ej. de uso:

```
int[] a = {1,2,3};
int[] b;
b = CopiaArrayPro(a);
```

33. Escribe una función “InsertaEnArray” a la que le pasas tres parámetros: un array de enteros, un valor entero y una posición. La función insertará el valor en la posición indicada, desplazando el resto de valores para hacerle hueco. Ej.: Si tenemos el array [1,2,3,4,5] y queremos insertar el valor “26” en la posición “2”, el resultado será: [1,2,26,3,4,5]. *Nota: el array habrá que pasarlo por referencia; esto es necesario siempre que hay que modificar el tamaño de un array.*

34. Escribe una función “BorraDeArray” a la que le pasas dos parámetros: un array de enteros y una posición. La función eliminará el elemento colocado en la posición indicada. Ej.: Si tenemos el array [5,7,2,8,1] y queremos eliminar la posición 1, el resultado será: [5,2,8,1]. *Nota: el array habrá que pasarlo por referencia; esto es necesario siempre que hay que modificar el tamaño de un array.*

35. Escribe una función “Elimina1ElementoArray” a la que le pasas dos parámetros: un array de enteros y un valor entero. La función eliminará del array el valor entero independientemente de la posición en la que se encuentre. Si el valor se repite, se eliminará sólo la primera vez que aparece el valor. *Nota: el array habrá que pasarlo por referencia; esto es necesario siempre que hay que modificar el tamaño de un array.*

36. Escribe una función “EliminaElementosArray” a la que le pasas dos parámetros: un array de enteros y un valor entero. La función eliminará del array el valor entero independientemente de la posición en la que se encuentre. Si el valor se repite, se eliminarán todas las veces que se repite. *Nota: el array habrá que pasarlo por referencia; esto es necesario siempre que hay que modificar el tamaño de un array.*

37. Escribe una función “InsertaArrayEnArray” a la que le pasas tres parámetros: un array de enteros, una posición de ese array, y otro array de enteros. La función insertará en el primer array, a partir de la posición indicada, todo el contenido del segundo array. Ej.: [6, 2, 1, 3], posición: 2, insertar: [12, 13], resultado = [6, 2, 12, 13, 1, 3]. *Nota: el*

array habrá que pasarlo por referencia; esto es necesario siempre que hay que modificar el tamaño de un array.

38. Escribe una función "SubArray" a la que le pasas un array y dos posiciones. La función te devuelve otro array que contiene los datos comprendidos entre ambas posiciones (incluidas ambas dos). Ej.: [3, 6, 2, 8, 9], posiciones, 1 y 3, resultado = [6, 2, 8].
39. Escribe una función "RecortaArray" a la que le pasas un array y dos posiciones. La función modificará el array inicial quedándose sólo con lo contenido entre ambas posiciones (como el ejercicio anterior pero en vez de devolver un array diferente modifica el array que le pasas). *Nota: el array habrá que pasarlo por referencia; esto es necesario siempre que hay que modificar el tamaño de un array.*
40. Escribe la función "OrdenaSelección" que realiza la ordenación de un array mediante el algoritmo de selección. El algoritmo funciona de la siguiente forma: cogemos el array que queremos ordenar y un array vacío; quitamos del primer array el elemento más pequeño y lo insertamos en el segundo array; repetimos el proceso hasta que el primer array esté vacío y el segundo contenga todos los elementos ordenados de menor a mayor.

Para realizar este ejercicio con mayor facilidad se recomienda crea las siguientes funciones (no es necesario, se puede hacer sin ellas):

- "InsertaAlFinalArray" que te inserta un elemento al final de un array (incrementando su tamaño).
- "PosMinArray" que te devuelve la posición donde se encuentra el elemento más pequeño del array.