

## Ejercicios de Arreglos

1. Escribe el valor final de las variables:

a)

```
float A[ 3];
int x;
for (int i=0; i<3; i++)
    A[i] = i*3;
x=A[0]+A[1]+A[2];
```

El valor de **x** es \_\_\_\_\_

El valor de **A[0]** es \_\_\_\_\_

El valor de **A[1]** es \_\_\_\_\_

El valor de **A[2]** es \_\_\_\_\_

b)

```
int B[5 ];
B[4]=1;
B[B[4]]=2;
B[B[B[4]]]=0;
B[B[B[B[4]]]]=3;
B[B[B[B[B[4]]]]]=4;
```

El valor de **B[0]** es \_\_\_\_\_

El valor de **B[1]** es \_\_\_\_\_

El valor de **B[2]** es \_\_\_\_\_

El valor de **B[3]** es \_\_\_\_\_

El valor de **B[4]** es \_\_\_\_\_

c)

```
int C[4 ];
int y;
for (int i=4 - 1; i>=0; i--)
    C[i] = i/2;
y=C[0]+C[1]+C[2]+C[3];
```

El valor de **y** es \_\_\_\_\_

El valor de **C[0]** es \_\_\_\_\_

El valor de **C[1]** es \_\_\_\_\_

El valor de **C[2]** es \_\_\_\_\_

El valor de **C[3]** es \_\_\_\_\_

2. Escribe el código en C de los siguientes problemas

a) Crear un arreglo de enteros y asignarle a cada localidad los valores 0,3,6,...,3\*n donde n es la longitud del arreglo.

Por ejemplo, si definimos el siguiente arreglo:

```
int A[ 6];
```

Y después ejecutamos la instrucción...

...las localidades de A deben ser  $A[0]=0$ ,  $A[1]=3$ ,  $A[2]=6$ ,  $A[3]=9$ ,  $A[4]=12$  y  $A[5]=15$ .

- b) Leer o generar de forma aleatoria un arreglo de enteros y determinar el número de ceros que se encuentran dentro del arreglo.
  - c) Leer o generar de forma aleatoria un arreglo de enteros y calcular el resultado de sumar sólo los números pares que hay dentro del arreglo.
  - d) Leer o generar de forma aleatoria un arreglo de enteros y un valor entero x. el programa debe calcular el número de veces que se repite el valor de x en el arreglo.
  - e) Leer o generar de forma aleatoria un arreglo de enteros y dos valores enteros old y new. El programa debe reemplazar todos los valores old del arreglo por el valor de new.
- 3) Escribe el código en C de un programa que lea o genere de forma aleatoria un arreglo de enteros y dos valores enteros que corresponden a dos localidades del arreglo. El programa intercambia los valores de las localidades i y j del arreglo. Por ejemplo, si el arreglo A tuviera los valores  $A = [8\ 1\ 4\ 7\ 5\ 10\ 6\ 3\ 9]$ , la operación **que** intercambia los valores  $A[2]$  y  $A[5]$ , modificando el arreglo de la siguiente forma  $A = [8\ 1\ 10\ 7\ 5\ 4\ 6\ 3\ 9]$ .
- 4) Escribe el código en C de un programa que lea o genere de forma aleatoria un arreglo de enteros. El programa invierte la secuencia de valores del arreglo. Por ejemplo, si el arreglo A tuviera los valores  $A = [10\ 3\ 5\ 2\ 3\ 9\ 10\ 4\ 7\ 1]$ , la operación intercambia los valores del arreglo invirtiendo la secuencia del arreglo:  $A = [1\ 7\ 4\ 10\ 9\ 3\ 2\ 5\ 3\ 10]$ .
- 5). Escribe el código en C de un programa que lea o genere de forma aleatoria un arreglo de enteros. El programa mueve a cada elemento una posición adelante, colocando el último valor del arreglo en la primera localidad del arreglo resultante. Por ejemplo, si el arreglo A tuviera los valores  $A = [0\ 1\ 2\ 3\ 4\ 5\ 6\ 7]$ , después de evaluar el arreglo se modifica de la siguiente manera:  $A = [7\ 0\ 1\ 2\ 3\ 4\ 5\ 6]$ .
- 6) Escribe el código en C de un programa que lea o genere de forma aleatoria un arreglo de enteros y dos valores enteros inf y sup. El programa debe calcular **mayor valor** del arreglo dentro del rango definido por inf y sup. Por ejemplo, si tenemos el siguiente arreglo:  $A = [10\ 12\ 2\ 0\ 4\ 35\ 24]$ , y los valores 0 y 3 para inf y sup respectivamente,regresa **12**.