

AEDD - Guía Práctica 6: Arreglos

Consideraciones básicas para arreglos:

- Los **elementos** del arreglo son de un **único tipo**
- **Siempre** tienen un **tamaño físico**
- **Pueden** manejar un **tamaño lógico**

Ejercicios propuestos:

[Tarea "Arreglos Unidimensionales \(Vectores\)" del juez en línea](#)

1. Declarar un arreglo de N componentes numéricos enteros. Luego, implementar variantes para la inicialización de sus componentes:
 - Asignando valores iniciales por extensión en la declaración.
 - Definiendo la función void inicializar_teclado (int V[], int tam) que carga los valores a través de la entrada estándar (teclado).
 - Definiendo la función void inicializar_aleatorio(int V[], int tam) que carga los valores a través de funciones de **generación de números aleatorios** haciendo uso de las funciones **srand(time(NULL))** y **rand()**.
2. Implementar la función void imprimir_vector(int V[], int tam) que permita visualizar los valores del vector V a través de la salida estándar (pantalla).
3. Crea una función que tome un arreglo de números como parámetro, encuentre el elemento **mayor y lo devuelva**.
4. Escribe una función que tome un arreglo de números como parámetro, **calcule la suma y el promedio de los elementos y los devuelva**.
5. Crea una función que tome un arreglo de números como parámetro y **devuelva un arreglo con los elementos en orden inverso**.
6. **Escribe una función que tome un arreglo de números y un número específico como parámetro, y devuelva el número de veces que aparece en el arreglo.**
7. **Crea una función bool que tome un arreglo de números como parámetro y determine si todos los elementos son iguales.**
8. **Escribir la función int mayor(int A[], int inf, int sup), que recibe un arreglo de N enteros y dos valores enteros inf y sup. La función regresa el mayor valor del arreglo dentro del rango definido por inf y sup. Se asume que inf >= 0, que sup < N y que inf <= sup.**

Ejemplo:

Sea el arreglo: A = [10 12 2 0 4 35 24], la instrucción mayor(A, 0, 3) regresa 12 y mayor(A, 0, 6) regresa 35.

9. **Para el vector X de N elementos numéricos reales, informar el mismo vector pero normalizado, es decir cada componente X[i] se debe reemplazar por:**

$$\frac{(X[i] - \text{Min}) * 100}{\text{Max} - \text{Min}} \text{ donde } \text{Min} = \min\{X[]\} \text{ y } \text{Max} = \max\{X[]\}$$

10. Se leen 10 valores enteros menores que 30. Luego se leen valores enteros positivos hasta que la suma de los dígitos de uno de los valores leídos, sea igual a alguno de los 10 valores inicialmente leídos. Informar el número que cumplió esta condición.

Ejemplo:

```
Valores iniciales: 1 6 25 18 23 2 6 19 14 13
i.    9425
ii.   340
iii.  694 Imprimir: 694
```

11. Escribir un programa que lea caracteres (letras minúsculas) por teclado, hasta Ctr+Z. Se debe mostrar la frecuencia de aparición de cada letra.

Ejemplo: Lee caracteres hasta que se presiona CTRL+Z.

```
int x;
...
cout << "Ingresar x (CTRL + Z para terminar)";
while(cin >> x){
    ...
    cout << "Ingresar x (CTRL + Z para terminar)";
}
...
```

12. Escribir la función bool todos_distintos(int vector[]) que indique si todos los elementos de una lista de N elementos son diferentes.

Ejemplo:

```
todos_distintos([6, 6, 6]) → False
todos_distintos([6, 6, 1]) → False
todos_distintos([0, 90, 1]) → True
```

13. Escribir una función que permita intercambiar dos elementos cualesquiera dentro del vector de N enteros no repetidos, dando como parámetros de entrada los elementos a intercambiar (verificar primero si los elementos están contenidos en el vector para poder intercambiarlos).

Ejemplo:

```
Intercambiar([3,5,4,6,7],3,6) → [6,5,4,3,7]
```

14. Escribir una función que permita rotar una posición a la izquierda todos los elementos del vector. La rotación es circular (el elemento en la posición cero se desplaza "circularmente" a la última posición del vector).

Ejemplo:

```
rotar_izquierda([1,2,3,4]) → [2,3,4,1]
```

15. Escribir una función que permita rotar circularmente una posición a la derecha todos los elementos del vector.

Ejemplo:

```
rotar_derecha([1,2,3,4]) → [4,1,2,3]
```

16. Escribir una función denominada Quini 6, que reciba un número de jugador y 2 arreglos no ordenados de tamaño 6: en uno están los números elegidos por el jugador y en el otro los números que salieron sorteados. Se eligen seis números distintos del 00 al 45. La función debe imprimir un mensaje con el número del jugador en caso de que sea ganador.