

AEDD - Guía Práctica 8: Arreglos unidimensionales

Consideraciones básicas para arreglos:

- ★ Son de un único tipo
- ★ Siempre tienen un tamaño físico
- ★ Pueden manejar un tamaño lógico

Ejercicios introductorios

Declarar un arreglo de **N** componentes numéricos enteros y luego:

- Implementar variantes para la inicialización de sus componentes:
 - a) Asignando valores iniciales por extensión en la declaración.
 - b) `int inicializar_teclado(int V[], int tam):` Ingresando los valores a través de la entrada estándar (teclado).
 - c) `int inicializar_aleatorio(int V[], int tam):` Asignando valores a través de funciones de generación de números aleatorios haciendo uso de las funciones `srand(time(NULL))` y `rand()`.
- Implementar la función `int imprimir_vector(int V[], int tam)` que permita visualizar los valores del vector V a través de la salida estándar (pantalla).

Implementación de arreglos

Para la resolución de los problemas, se asume que los arreglos están previamente cargados.

Problema 1: 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 $inf \geq 0$, que $sup < N$ y que $inf \leq sup$.

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.

Problema 2: 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:

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

Problema 3: 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
```

Problema 4: Escribir un programa que reciba caracteres (letras minúsculas) por teclado, hasta Ctr+Z. Se debe mostrar la frecuencia de aparición de cada letra.



Ctr+Z (fin de archivo) El operador de extracción de flujo es >>. Cuando este encuentra el fin de archivo en una entrada, devuelve cero (false), de lo contrario devuelve una referencia al objeto mediante el cual es llamado (cin).

Ejemplo: Lee números 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)";
}
```

Problema 5: Escribir la función `bool todos_iguales(int vector[])` que indique si todos los elementos de una lista de N elementos son iguales.

Ejemplo:

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

Problema 6: 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]`

Problema 7: Movimiento de elementos en un vector

7.a) Escribir una función que permita rotar una posición a la izquierda todos los elementos del vector, colocando el elemento que sale de la posición cero en la última posición del vector.

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

7.b) Escribir una función que permita rotar una posición a la derecha todos los elementos del vector, colocando el elemento que sale de la última posición en la posición cero del vector.

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

Problema 8: 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 que sea ganador.



Implementación de listas con arreglos

Problema 9: Una función debe recibir un vector de números enteros y su tamaño lógico. Debe retornar los datos actualizados, insertando un elemento detrás de cada elemento que tenga un valor X múltiplo de 100, con el valor $X+1$; y además devolver la cantidad de elementos insertados.

Suponer que el tamaño físico del arreglo alcanza para hacer las inserciones. No recorrer el arreglo más de una vez.

Problema 10: Una función debe recibir un vector **ordenado** de números enteros positivos y su tamaño lógico. Debe retornar los datos actualizados, eliminando del vector los valores repetidos; y además devolver la cantidad de elementos pares eliminados y el promedio de los valores que quedan en el vector.

Problema 11: Una función debe recibir un vector ordenado de números enteros positivos y su tamaño lógico. Debe retornar los datos actualizados, eliminando del vector todos los valores impares que se encuentran entre 2 valores pares adyacentes; y además devolver la cantidad de elementos impares eliminados y la mayor diferencia entre dos elementos pares sucesivos de los valores que quedan en el vector.

Escriba el prototipo adecuado para esta función y su definición.

Ejemplo:

```
V = {1, 2, 3, 4, 7, 9, 12, 14, 15, 20}
```

```
T = 10
```

```
luego de la llamada a la función
```

```
V = {1, 2, 4, 7, 9, 12, 14, 20}
```

```
T = 8
```

```
Cantidad de elementos impares eliminados: 2
```

```
Mayor diferencia entre dos pares sucesivos: 6
```

Problema 12: La función llama `elimOrdenar` recibe como parámetros un arreglo `A` de 500 números enteros, el tamaño lógico, `TLA`, del arreglo y un valor entero `X`. El arreglo `A` está desordenado. La función deberá devolver el vector `A` actualizado de la siguiente forma:

- Se deben eliminar todos los valores que sean múltiplos de `X`.
- El vector debe quedar ordenado ascendentemente

La función deberá retornar la cantidad de elementos eliminados de `A`.

Problema 13: La función ***Uno()*** recibe dos arreglos de 500 números enteros (`A` y `B`), los tamaños lógicos de cada uno (`TA` y `TB`, `TB <= TA`) y un valor booleano (`C`). El arreglo `A` y el arreglo `B` están desordenados.

La función ***Uno()*** debe retornar un vector `V` y su tamaño lógico `TV`, ordenado ascendentemente si `C==True` (descendentemente en otro caso) conteniendo solamente los valores de `B` que se encuentran en `A`, sin repeticiones. Los vectores `A` y `B` pueden contener elementos repetidos.