

**GUIA N04****Arrays y Matrices**

---

**Arrays Unidimensionales**

- 1) Calcular la cantidad de números pares presentes en un arreglo A de n números enteros.
- 2) Determinar el máximo elemento y el mínimo elemento de un arreglo A de n números.
- 3) Sean A y B dos arreglos de N números. Se dice que:
  - $A = B$  si  $\forall i = \underline{1, n}$  el i-ésimo elemento de A es igual al i-ésimo elemento de B.
  - $A < B$  si  $\forall i = \underline{1, n}$  el i-ésimo elemento de A es menor al i-ésimo elemento de B.
  - $A > B$  si  $\forall i = \underline{1, n}$  el i-ésimo elemento de A es mayor al i-ésimo elemento de B.Escriba un algoritmo que retorne:
  - el valor 1 si  $A = B$
  - el valor 2 si  $A < B$
  - el valor 3 si  $A > B$
  - el valor 0 si no se cumple ninguna de las condiciones anteriores.
- 4) El videojuego Peruvian Fighter tiene una dinámica de juego bastante sencilla. Se enfrentan 2 jugadores, cada uno de ellos elige un conjunto de personajes y forma un equipo, cada personaje tiene un determinado poder de pelea. Los encuentros se realizan en orden, es decir, el primer personaje de jugador1 se enfrenta con el primer personaje de jugador2, luego se enfrentan los segundos y así sucesivamente. Debe tomar en cuenta que tanto el personaje que gana como el que pierde deben salir y dar paso a los siguientes. El jugador que tenga una mayor cantidad de encuentros ganados se quedará con la partida.

Implemente un programa que reciba 2 arreglos, cada uno de ellos con el poder de pelea de los personajes elegidos por cada jugador y pueda “predecir” quién será el ganador. En principio, si un personaje tiene mayor poder de pelea que otro, es altamente probable que gane la partida, por lo que nuestra predicción seguirá esa lógica.

### Ejemplo:

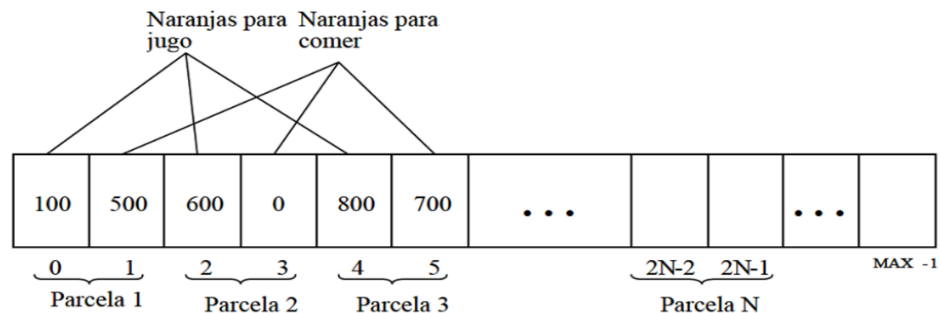
poderes\_j1 = {100, 150, 220}

poderes\_j2 = {90, 240, 624}

Salida: Jugador2 sería el ganador.

En este caso, la primera partida la ganaría jugador1 ( $100 > 90$ ), la segunda y la tercera serían ganadas por jugador2, por lo que el programa declara ganador a jugador2.

- 5) A partir de un arreglo llamado **X** de 50 elementos enteros, crear y mostrar:
- Otro arreglo **DX** tal que el primer elemento sea la diferencia entre el segundo y el primer elemento de **X**, el segundo elemento de **DX** sea la diferencia entre el tercero y el segundo de **X** y así sucesivamente.
  - Idéntico al apartado (a) pero sobre sí mismo, o sea sin generar un arreglo **DX**. Desarrolle un algoritmo para cada caso.
- 6) Una cooperativa de productores de naranjas almacena el total de toneladas cosechadas durante el último año en **N** parcelas ( $1 \leq N \leq 50$ ). En cada parcela se pueden cultivar dos tipos de naranjas: para jugo y para comer. Se conoce el total de toneladas cosechadas de cada uno de los tipos de naranjas. Si en una parcela no se hubiera cosechado alguno de los tipos, entonces habrá 0. La información se almacena en un arreglo como se muestra en el siguiente ejemplo:



- o En la parcela 1 se cosecharon: 100 toneladas de naranjas para jugo y 500 toneladas de naranjas para comer.
- o En la parcela 2 se cosecharon: 600 toneladas de naranjas para jugo y 0 toneladas de naranjas para comer.

NOTA: Observe que la información de una misma parcela ocupa posiciones consecutivas en el arreglo.

Escriba un algoritmo que permita realizar las siguientes tareas:

- Calcular e imprimir el total de la producción por parcela.

- b)** Eliminar la información de una parcela. El dato requerido para esta opción es el número de parcela a eliminar.
- c)** Buscar e imprimir el número de una parcela (si hubiera) que no haya tenido producción de ninguno de los tipos de naranjas. Es decir, durante el último año su producción total fue 0.