

# Contenido de la actividad

Entregar la resolución de los siguientes ejercicios:

Escribir un programa efectúe las siguientes operaciones:

- a) Declarar un arreglo de 20 posiciones de tipo int.
  - b) Cargar sus elementos y mostrarlo en funciones (utilizando punteros).
  - c) Realizar una función que cuente la cantidad de números pares que se cargaron en el arreglo y acumular los impares.
- Mostrar los resultados (recuerde que las funciones que calculan no imprimen).
- d) Recorrer con el puntero el arreglo, mostrando la dirección y el contenido de cada posición en forma prolija.

Responda:

- **a) ¿Qué valor representa arreglo[2]?**  
el tercer elemento (en posición) del arreglo. Accede al elemento de índice 2
- **b) ¿Qué valor representa \*(&arreglo[3+2])?**  
arreglo[3+2] sería lo mismo que decir arreglo[5], o sea índice 5, 6to elemento del array. Luego procede a tomar la dirección del arreglo con el ampersand, y deshace esto mismo con el asterisco al tomar el contenido de la dirección de memoria que obtuvimos. O sea, termina siendo equivalente a arreglo[5]
- **c) ¿Qué valor representa \*(arreglo+2)?**  
(arreglo+2) representa una operación de aritmética de punteros. Avanza desde el índice cero un espacio de dos elementos y queda en el tercer elemento, que sería el índice 2. Todo esto es a nivel de punteros, y en memoria el puntero avanza la cantidad de bytes que representa el tipo de dato al que está asociado. Luego accede al elemento de la posición 3 (índice 2) mediante el asterisco. Sería equivalente a la expresión: arreglo[2].
- **d) ¿Qué valor representa arreglo[0]?**  
Es el primer elemento, o sea la primera posición accedida del array. Si “arreglo” es un puntero al índice ‘0’ del array, arreglo[0] es el contenido de la primera posición del array. Si por ejemplo contenía un carácter, arreglo[0] podría ser ‘a’.
- **e) ¿Qué valor representa \*(arreglo-2)?**  
Esta expresión es peligrosa porque si no se tiene cuidado en donde uno está posicionado, puede terminar accediendo a un lugar de memoria inválido y tener resultados imprevisibles o no definidos. Si está usada correctamente, “arreglo” podría representar un subarray de un array más grande. “arreglo” apunta a cierta dirección de memoria, y arreglo-2 va dos posiciones para atrás desde la referencia que es “arreglo”. Luego procede a acceder al dato que está allí almacenado mediante el asterisco. En “restaAritmPunt.c” trato de ejemplificar esto.