

Práctico 3 – Trabajar con vectores

OBJETIVOS:

- 1) Aplicar metodologías avanzadas de programación y técnicas de resolución de problemas como partición y refinamiento.
- 2) Implementar estándares y buenas prácticas de programación, enfatizando la correcta especificación semántica y sintáctica.
- 3) Reforzar el uso de estructuras de control y desarrollar habilidades en la implementación detallada de algoritmos

En todos los casos se deberá indicar las PRE/POS condiciones y realizar corridas a mano.

Ejercicio 1

Implementar una función que reciba un array de enteros y retorne una cadena de texto con su contenido formateado.

Firma: `public static String mostrarv(int []v);`

Resultado esperado para vector [6,3,5,1,8,7,2,4]: "6 - 3 - 5 - 1 - 8 - 7 - 2 - 4"

Ejercicio 2

Implementar una función que calcule el promedio de los valores contenidos en un array de enteros.

Firma: `public static double promedio(int []v);`

Ejercicio 3

Implementar una función que retorne una cadena con los valores impares del array.

Firma: `public static String nuestroValoresImpares(int v[]);`

Ejercicio 4

Implementar una función que retorne los valores ubicados en las posiciones pares del array.

Firma: `public static String nuestroPosPares(int v[]);`

Ejercicio 5

Implementar una función que retorne el valor máximo de un array.

Firma: `public int maxVec(int []v);`

Considerar los siguientes 2 casos:

- Caso 1: se asume que el array no está vacío y no esta ordenado
- Caso 2: se asume que el array no está vacío y esta ordenado en forma ascendente.

Ejercicio 6

Implementar una función que determine si un array es simétrico (es decir, si se lee igual de izquierda a derecha que de derecha a izquierda).

Firma: public static boolean esSimetrico(int[] v);

Entrada: [1, 2, 3, 2, 1] Salida: true

Ejercicio 7

Implementar una función que retorne la posición del menor valor dentro de un rango específico del array (posiciones dadas).

Firma: public static int posMinVec(int []v,int posDesde, int posHasta);

Ejercicio 8

Implementar una función que reciba un array y un valor entero, retornando true si el valor existe en el array y false en caso contrario.

Firma: public static boolean buscarVec(int []v, int elemento);

Considerar los siguientes casos:

- Caso 1: el vector no está vacío y no esta ordenado
- Caso 2: el vector no está vacío y se encuentra ordenado en forma ascendente

Ejercicio 9

Dado dos vectores v1 y v2, ambos ordenados en forma ascendente, implementar una función que retorne un nuevo vector ordenado que contenga los elementos de ambos.

Firma: public static int [] unirVectoresOrdenados(int []v1, int []v2);

Ejercicio 10

Implementar una función que invierta el orden de los elementos de un array.

Firma: public static void invertirVector(int[] v);

Entrada: [1, 2, 3, 4, 5] Salida: [5, 4, 3, 2, 1]

Ejercicio 11

Implementar una función que elimine todas las apariciones de un valor específico en el array y retorne un nuevo array con los elementos restantes.

Entrada: v = [3, 4, 5, 3, 7], elemento = 3 Salida: [4, 5, 7]

Ejercicio 12

Implementar un algoritmo de ordenamiento ascendente utilizando la función obtenerPosMinima implementada en el ejercicio 7.

Firma: public static void ordenarvec(int []v);