

ACTIVIDADES ESTRUCTURAS DE ALMACENAMIENTO

Actividad 1

Escribir un programa que lea 10 números de tipo entero y los almacene en un vector. Implementar una función que muestre el contenido del vector indicando para cada elemento su posición.

Actividad 2

Escribir un programa con implemente las siguientes funciones para un vector de 10 posiciones de tipo entero.

- void leerVector(int v[]) lee 10 números de tipo entero y los almacene en un vector
- Implementar una función que muestre el contenido del vector indicando para cada elemento su posición **en orden inverso**.

Actividad 2

Implementar un programa con las siguientes funciones:

- void leerVector(int v[]) : función que lee los elementos de un vector
- int suma(int v[]) retorna la suma de los elementos de un vector
- int max(int v[]) retorna el máximo de un vector
- int sumaNumerosPares(int v[]) retorna la suma de los valores pares de un vector
- int sumaPosicionesImpares(int v[]) retorna la suma de los valores almacenados en las posiciones impares del vector.

Actividad 3

Leer las dimensiones de una matriz de enteros por teclado, y rellenarla con valores aleatorios entre 1 y 10

Nota: para generar valores aleatorios entre 1 y 10 `Math.random()*10+1`

Actividad 4

Calcular la varianza de un vector. Se define la varianza de un vector, como la suma de los cuadrados de cada elementos dividido entre el numero de elementos

Actividad 5

Calcular la mediana de un vector. Es aquella que deja tantos menores como mayores. solución ordenar y retornar la posición central

Actividad 6

Implementar una función que realice la búsqueda secuencial de un valor sobre un vector de reales que recibe como parámetro. La cabecera de la función `int búsqueda(float v[], float value)`

Actividad 7

Dado un vector de números reales, escriba un método que ordene los elementos del vector de tal forma que los números pares aparezcan antes que los números impares. Además, los números pares deberán estar ordenados de forma ascendente, mientras que los números impares deberán estar ordenados de forma descendente. Esto es, el vector `{1,2,3,4,5,6}` quedará como `{2,4,6,5,3,1}`.

Actividad 8

En Java, para generar números pseudoaleatorios, se puede utilizar la función `Math.random()` definida en la clase `java.lang.Math`. Dicha función genera una secuencia de números pseudoaleatorios que se supone sigue una distribución uniforme (esto es, todos los valores aparecerán con la misma probabilidad). Escriba un programa que compruebe si el generador de números pseudoaleatorios de Java genera realmente números aleatorios con una distribución uniforme. Sugerencia: Genere un gran número de números aleatorios (entre 0 y

100, por ejemplo) comprendidos entre 1 y 10 y compruebe que la distribución resultante del número de veces que aparece cada número es (más o menos) uniforme. Para ello, una vez calculado el vector, realizaremos el conteo de cuantas veces aparece cada valor.

Una vez realizado el conteo, calcular la media y ver si la dispersión de los valores es compacta.

Actividad 9.

Se dice que un vector tiene un pico, si dada una posición i los valores de las posiciones -1 y $i+1$ son menores que ella.



Se dice que un vector tiene un valle cuando dada una posición i los valores de las posiciones -1 y $+1$ son mayores que ella.

Implementar una función que determine si tiene pico y una función que determine si un vector tiene valle.

Actividad 10

Implementar una función que retorne el numero de veces que aparece un valor dentro del vector.

```
int buscarValor(int v[], int valor)
```

Generar el vector con elementos aleatorios entre 1 y 10