

ACTIVIDADES ESTRUCTURAS DE ALMACENAMIENTO (REPASO)

Ejercicio 1

Implementar un método insertaValor, que recibe como parámetros:

- Un vector reales
- Un valor real
- Una posición como valor entero

Inserta el valor en la posición indicada. Retorna un vector con el valor incluido.

Nota: Insertar un elemento en el vector desplaza una posición hacia la derecha a los elementos del vector que han de quedar detrás del elemento insertado. Además, la inserción ocasiona la “desaparición” del último elemento del vector.

Ejercicio 2

Implemente un método llamado secuencia que realice la búsqueda de la secuencia en orden creciente más larga dentro de un vector de enteros. El método ha de mostrar tanto la posición de la primera componente de la secuencia como el tamaño de la misma.

Ejercicio 3

Una cadena de ADN se representa como una secuencia circular de bases (adenina, timina, citosina y guanina) que es única para cada ser vivo, por ejemplo:

A	T	G
T		C
A	T	G

Dicha cadena se puede representar como un vector de caracteres recorriéndola en sentido horario desde la parte superior izquierda:

A	T	G	C	G	T	A	T
---	---	---	---	---	---	---	---

Se pide diseñar un método booleano que nos devuelva true si dos cadenas de ADN coinciden. MUY IMPORTANTE: La secuencia de ADN es cíclica, por lo que puede comenzar en cualquier posición. Por ejemplo, las dos secuencias siguientes coinciden:

A	T	G	C	G	T	A	T
---	---	---	---	---	---	---	---

A	T	A	T	G	C	G	T
---	---	---	---	---	---	---	---

Ejercicio 4

Crear un programa modular para jugar a las 7 y media. Se trata de un juego de cartas (con baraja española) en el que el objetivo es alcanzar una puntuación de 7.5. Cada carta del 1 al 7 tiene su valor nominal y cada figura (sota, caballo y rey) vale 0.5 puntos.

Para barajar, mezcle los elementos de un vector de cartas intercambiando en repetidas ocasiones cartas elegidas al azar con la ayuda de la función `Math.random()`

Cada jugador almacena en un vector las cartas elegidas

Implementar las siguientes funciones:

- `int v[] generarBaraja()`
- `void mezclarBaraja(int bajara[])`
- `char elegirCarta(int pos)` : retorna un carácter que representa la carta.
- `int calcularPuntuación(int jugada[])`

Ejercicio 5

Desarrollar un método para rellenar una matriz A $[n \times m]$ de tal forma que contenga en las filas pares 10 y en las columnas impares 11, el resto 0

	0	1	2	3	4	5	6
0	10	10	10	10	10	10	10
1	0	11	0	11	0	11	0
2	10	10	10	10	10	10	10
3	0	11	0	11	0	11	0
4	10	10	10	10	10	10	10
5	0	11	0	11	0	11	0
6	10	10	10	10	10	10	10

Ejercicio 6

Realizar un programa que ingrese las notas de n estudiantes en m asignaturas. Obtener la media de los n estudiantes y almacenarlos en un vector

	Asignatura 1	Asignatura 2	Asignatura 3
Estudiante 1	4	7	8
Estudiante 2	2	9	1
Estudiante n	5	3	6

Promedio
6,33
4,00
4,67

- método para mostrar el estudiante con mayor y menor media
- método para mostrar el estudiante con la menor nota obtenida en cualquier asignatura
- método para mostrar el estudiante con mayor nota en la asignatura 3
- método para mostrar la media de la asignatura 1

Ejercicio 7

Determinar si una matriz de nxn es mágica. Una matriz es mágica si la suma de cada fila, columna y diagonales es la misma.

Implementar las siguientes funciones

- int sumaFila(int m[][], int fila)
- int sumaColumna(int m[][], int columna)
- int sumaDiagonalPrincipal(int m[][])
- int sumaDiagonalSecundaria(int m[][])