



## PROGRAMACIÓN 1º CFGS DAW Ejercicios tema 4 - Vectores



- 1º** Crea un vector de diez números enteros. Inicialízala todas las casillas a -1. Mostrar el vector resultante. Diseñalo modularmente.
- 2º** Crea dos vectores; uno que tenga los primeros 10 números pares y otro que tenga los 10 primeros impares. Modularmente.
- 3º** Escribe un programa modular que pida diez valores reales y los almacene en un vector. Recórrelo luego para averiguar la media de dichos números.
- 4º** Crea un programa modular que inicie un vector con diez números aleatorios. Después un módulo me indicará cual es el valor máximo y la posición que ocupa. Caso de haber más de un máximo me debe devolver la posición del último.
- 5º** Diseña un programa que genere un vector con números al azar que oscilan entre [-100 y 100]. Después realiza un módulo que indique cuantos números positivos y cuantos negativos hay.
- 6º** Escribe un programa que inicie un vector con números al azar de 1 a 100. Después crearemos un módulo al que le pasemos un número y me devuelva cuantas veces aparece ese número en el vector.
- 7º** Repetir el ejercicio anterior pero ahora, después de rellenar el vector, se le pasará un número al módulo y debe devolver la posición de la primera aparición de ese número en el vector; devolverá -1 si ese número no está en el vector.
- 8º** Dado un array de números de 5 posiciones con los siguiente valores {1,2,3,4,5}, guardar los valores de este array en otro array distinto pero con los valores invertidos, es decir, que el segundo array deberá tener los valores {5,4,3,2,1}.
- 9º** Crea una aplicación que pida un numero por teclado y después comprobaremos si el numero introducido es capicúa, es decir, que se lee igual sin importar la dirección. Por ejemplo, si introducimos 30303 es capicúa, si introducimos 30430 no es capicúa. Utiliza vectores para resolver el problema.
- 10º** Crea un programa que genere una combinación de la lotería primitiva cuidando de no repetir ningún número.
- 11º** Diseña un programa modular que inicie un vector de 20 posiciones con números al azar del 1 al 10. Después pide otro vector de tres posiciones por teclado. Crea un módulo al que le pasemos ambos vectores y nos calcule la posición en la que existe alguna repetición del vector más pequeño en el primero.



## PROGRAMACIÓN 1º CFGS DAW Ejercicios tema 4 - Vectores



### 12º Suma de vectores.

Realiza un programa que inicie dos vectores de 3 elementos y los sume en un tercero.

Ejemplo:

V1: 

|   |    |   |
|---|----|---|
| 2 | -3 | 4 |
|---|----|---|

V2: 

|   |   |   |
|---|---|---|
| 4 | 9 | 5 |
|---|---|---|

Suma: 

|   |   |   |
|---|---|---|
| 6 | 6 | 9 |
|---|---|---|

### 13º Iniciar respecto a una posición.

Realiza un programa que pida un valor entero, lo coloque en la primera posición de un vector e inicie el resto de elementos sumando uno al anterior.

Ejemplo, si introducimos un 9 en la primera posición el resto de componentes quedaría:

|   |    |    |    |    |    |    |
|---|----|----|----|----|----|----|
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|----|----|----|----|----|----|

Repite el ejercicio en otro módulo de forma que se inicie el primer elemento y el resto de componentes del vector sean el anterior más el índice actual. Ejemplo, si introducimos un 2 el primero el resto quedaría:

|   |   |   |   |    |    |    |
|---|---|---|---|----|----|----|
| 2 | 3 | 5 | 8 | 12 | 17 | 23 |
|---|---|---|---|----|----|----|

### 14º Donde está la mosca.

Vamos a intentar cazar una mosca. La mosca será un valor que se introduce en una posición de un vector; el jugador no ve el panel pero si ve las casillas a las que puede golpear. Si la mosca está en esa posición se acaba el juego, mosca cazada. Si la mosca no está en esa posición pueden ocurrir dos cosas: que la mosca esté en casillas adyacentes en cuyo caso la mosca revolotea y se sitúa en otra casilla o que la mosca no esté en casillas adyacentes, en este caso la mosca permanece donde está.

### 15º Juego de las parejas.

Realizaremos el juego de las parejas. Se inicia un vector de n casillas (siendo n un número par) y se colocan al azar parejas de números. Ese panel se oculta al jugador al que se le mostrará un panel vacío del que irá destapando de 2 en 2. Si los números destapados coinciden se quedan visibles si no se muestran un segundo y luego se ocultan. El jugador tratará de recordar qué números eran para encontrar a su pareja.

### 16º El río y la piedra.

Simularemos el lanzamiento de una piedra a un río. Se pedirá al usuario donde quiere lanzar la piedra (posición del vector) y la intensidad de la piedra (un número entero menor o igual que la dimensión máxima del vector). Cuando se lance, se almacena en



## PROGRAMACIÓN 1º CFGS DAW Ejercicios tema 4 - Vectores



esa casilla la *pedrá* y en las adyacentes se irán simulando las ondas con números que se van decrementando. Ejemplo, si damos una pedrá del 4 en la posición 6:

|   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 2 | 3 | 4 | 3 | 2 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

Después, cada segundo, se irá calmando el río restándole uno a las posiciones con número. La simulación se parará cuando el río vuelva a estar en calma; todo a cero.

### 17º Barquitos en río.

Realiza el juego de los barquitos pero con un vector. Sólo se pondrán submarinos (barcos de una casilla).

El juego consiste en lo siguiente:

- Los paneles serán de 20 posiciones.
- Se pondrán 4 barcos: el ordenador coloca los suyos al azar y el jugador humano es preguntado. Los barcos no pueden estar colocados de forma adyacente.
- Después, por turnos tira cada uno de los jugadores. El ordenador al azar y el humano es preguntado.
- El juego termina cuando uno de los dos acaba con la flota del otro.

Lógicamente se llevarán dos paneles para cada jugador: uno para su flota y otro para sus tiradas. Se irá mostrando el de sus tiradas. Al final del juego se muestran todos los paneles.

### 18º Buscaminas.

Realizaremos el juego del buscaminas con un vector. Para aquellos que no hayan jugado nunca (ni siquiera mientras estoy explicando algo) os recuerdo que el juego consiste en destapar todas las casillas de un vector menos las minas; si pisamos una mina el juego acaba y hemos perdido.

El juego nos proporcionará pistas, de forma que si destapamos una casilla y no hay una mina, esta casilla nos indicará cuantas minas hay adyacentes a esa posición.

Por lo tanto el ordenador debe preparar un panel de 20 casillas para nosotros en el que colocará 6 minas y las pistas correspondientes.

Ejemplo:

|   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | * | 1 | 1 | * | * | 2 | * | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

Este panel permanecerá oculto y es el jugador el que debe tratar de descubrirlo.