

ESTRUCTURA DE ALMACENAMIENTO - ARRAYS

Programación

EJERCICIOS

1. Escribe un programa que pida al usuario 4 números, los memorice (utilizando un array), calcule su media aritmética y después muestre en pantalla la media y los datos tecleados.
2. Escribe un programa que lea 10 números por teclado y que luego los muestre en orden inverso, es decir, el primero que se introduce es el último en mostrarse y viceversa.
3. Define tres arrays de 20 números enteros cada una, con nombres numero, cuadrado y cubo. Carga el array numero con valores aleatorios entre 0 y 100. En el array cuadrado se deben almacenar los cuadrados de los valores que hay en el array numero. En el array cubo se deben almacenar los cubos de los valores que hay en numero. A continuación, muestra el contenido de los tres arrays dispuesto en tres columnas.
4. Escribe un programa que almacene en una tabla el número de días que tiene cada mes (supondremos que es un año no bisiesto), pida al usuario que le indique un mes (1=enero, 12=diciembre) y muestre en pantalla el número de días que tiene ese mes.
5. Calcula la letra de un DNI, pediremos el DNI por teclado y nos devolverá el DNI completo. Para calcular la letra, cogeremos el resto de dividir nuestro dni entre 23, el resultado debe estar entre 0 y 22. Haz un método donde según el resultado

de la anterior formula busque en un array de caracteres la posición que corresponda a la letra. Esta es la tabla de caracteres:

Posición	Letra	Posición	Letra	Posición	Letra	Posición	Letra	Posición	Letra	Posición	Letra
0	T	4	G	8	P	12	N	16	Q	20	C
1	R	5	M	9	D	13	J	17	V	21	K
2	W	6	Y	10	X	14	Z	18	H	22	E
3	A	7	F	11	B	15	S	19	L		

6. Escribe un programa que genere 20 números enteros aleatorios entre 0 y 100 y que los almacene en un array. El programa debe ser capaz de pasar todos los números pares a las primeras posiciones del array (del 0 en adelante) y todos los números impares a las celdas restantes. Utiliza arrays auxiliares si es necesario.

7. Crea un programa que prepare espacio para un máximo de 10 nombres. Deberá mostrar al usuario un menú que le permita realizar las siguientes operaciones:
 1. Añadir un dato al final de los ya existentes.
 2. Insertar un dato en una cierta posición (los que queden detrás deberán desplazarse "a la derecha" para dejarle hueco; por ejemplo, si el array contiene "hola", "adios" y se pide insertar "bien" en la segunda posición, el array pasará a contener "hola", "bien", "adios".
 3. Borrar el dato que hay en una cierta posición (lo que estaban detrás deberán desplazarse "a la izquierda" para que no haya huecos; por ejemplo, si el array contiene "hola", "bien", "adios" y se pide borrar el dato de la segunda posición, el array pasará a contener "hola", "adios"
 4. Mostrar los datos que contiene el array.
 5. Salir del programa.

8. Crea el programa GESTISIMAL (GESTIÓN SIMplicada de Almacén) para llevar el control de los artículos de un almacén. De cada artículo se debe saber el código, la descripción, el precio de compra, el precio de venta y el stock (número de unidades). El menú del programa debe tener, al menos, las siguientes opciones:
 1. Listado
 2. Alta
 3. Baja
 4. Modificación
 5. Entrada de mercancía
 6. Salida de mercancía
 7. Salir

La entrada y salida de mercancía supone respectivamente el incremento y decremento de stock de un determinado artículo. Hay que controlar que no se pueda sacar más mercancía de la que hay en el almacén.

9. Hacer un programa que permita calcular el resultado de una votación de delegados en una clase. Crear una clase Persona que tenga como atributos nombre y numero de votos.

Será necesario utilizar un ARRAY de 24 posiciones para almacenar a los 24 alumnos que tiene la clase. El nombre se generará automáticamente como ALUMNO 1, ALUMNO 2, etc.

El programa tendrá que permitir realizar la votación a los 24 alumnos de la clase, de forma que no se pueden votar a si mismos, pueden votar únicamente a uno de sus compañeros o dejar el voto en blanco.

Cuando acaben de realizar las votaciones saldrá el nombre del delegado (el alumno/s que más votos tenga)

10. Desarrollar un programa que nos permita tener almacenados máximo 10 Pockemons. Los datos del pockemon que queremos almacenar son:

- Nombre (único, no pueden haber 2 pockemons que se llamen igual)
- Dueño (un mismo dueño puede tener varios pockemons)
- Poder (numérico)

En la clase Main crear un Array para almacenar los pockemons. Hacer los siguientes métodos:

- un método que permita insertar un pockemon en el Array.
- un método que permita mostrar todos las pockemon de un dueño determinado
 - o un método que permita modificar los datos de un pockemon introducido, sabiendo su nombre
- un método que permita eliminar un pockemon introduciendo su nombre

Para probar estos métodos desde la función principal:

- insertaremos 4 pockemons,
- Pediremos los datos de un dueño para ver cuantos pockemons tiene
- Pediremos el nombre de un pockemon para modificarlo
- Pediremos el nombre de un pockemon para borrarlo