

Objetivo de la práctica:

- Aprender el uso de funciones y la correcta modularización de programas.

COMENTARIOS:

Conceptos básicos: vectores, matrices y array multidimensionales.

Un *array* es una secuencia de objetos del mismo tipo. Los objetos se llaman elementos del *array* y se numeran consecutivamente 0, 1, 2.

El tipo de elementos almacenados en el array puede ser cualquier tipo de dato C++.

Adoptaremos el convenio de emplear **typedef** para crear este tipo de dato estructurado:

Vectores:

```
const int MAX = 20;

typedef int Vector20[MAX]; //define un vector de 20 enteros

Vector20 mi_vector; //declaro un vector de 20 enteros.
```

El acceso a los elementos de un vector se realiza mediante un índice. Los elementos están ordenados consecutivamente partiendo de 0. Ej. `mi_vector[3]=45;` //asigna el valor 45 a la cuarta componente

Matriz:

```
const int FIL = 10;
const int COL = 4;

typedef int Matriz10x4[FIL][COL]; Matriz10x4 mi_matriz;

mi_matriz[1][3]=32;
```

Array multidimensional:

```
const int FILA1 = 10;
...
const int FILAN =12;

typedef int Array_n_columnas[FILA1][FILA2]. . . .[FILAN];

Array_n_columnas mi_array_n_columnas;

mi_array_n_columnas [1][2] . . .[10] = 20;
```

Los arrays **se pasan por defecto por referencia** como argumentos de una función. La única diferencia con los tipos simples es que no se usa el ‘&’ ya que basta simplemente el nombre del array. Para pasar un array únicamente como parámetro de entrada se pone la palabra **const** delante del tipo del array.

PROBLEMAS

1. (**) Utilizando funciones implementar un programa que lea de teclado 10 números y realice con ellos las siguientes operaciones:
 - a) Calcule el max y min de ambos vectores.
 - b) Invierta el orden de los elementos.
 - c) Resetear el vector a un valor dado.
 - d) Leer 10 nuevos números por teclado e insertarlos en orden ascendente.

Todas estas opciones deben de presentarse a modo de menú.

2. (*) Escribir un programa que realice la suma de dos números enteros positivos muy grandes (máximo 200 cifras) y de el resultado exacto.

Nota: No se puede utilizar coma flotante, si se utiliza coma flotante el resultado es una aproximación.

Nota2: Pensad claramente la manera de guardar el número en el vector de manera que se facilite su suma.

3. (*) Realizar un programa que lea una matriz de tamaño 3x3 de números reales. El programa debe de realizar las siguientes operaciones:
 - a) Presentar por pantalla la matriz en formato tabular
A B C
D E F
G H Y
 - b) Calcular su traspuesta y presentarla a continuación.
 - c) Calcular la media de los elementos, el elemento máximo y el mínimo.
 - d) Representar la matriz como una concatenación de tres vectores en uno solo de tamaño 9.
 - e) Pedir por teclado una posición i , j y realizar una función que traslade dicha posición bidimensional a la del anterior vector, devolviendo el elemento i , j pero actuando sobre el vector concatenado.

4. (*) Realiza un programa que pida de teclado dos matrices de tamaño 3*3 de valores enteros y realice el producto matricial de ambas.

5. (**) Una de las grandes aplicaciones de las matrices es el tratamiento digital de imágenes. Dados 3 matrices de 5x5 que representan las tres componentes de color r,g,b de una reducida imagen de tamaño 5x5:

- a) Los valores de cada matriz se pedirán por teclado y sus valores deberán de estar en el rango [0..255] no permitiendo la entrada de cualquier otro valor.
- b) Realizar el proceso de suavizado de la imagen. Esta funcionalidad consistirá en obtener una nueva matriz que contenga en cada celda el valor de la celda original, sumada a los valores de sus vecinos más próximos (laterales y diagonales) dividido entre el número de vecinos + 1. En las celdas que limitan la matriz se realizará la

media con los vecinos que posea que será menor que 8 y se dividirá por el número de vecinos + 1.

6. (*) Realizar un programa que pida una matriz de tipo real cuyas dimensiones son $M \times N$ leídas desde teclado. Añadir al programa un subprograma que permita sumar el número de elementos positivos y negativos de una matriz.

Aclaración: La matriz tiene dimensión 10×10 , por teclado pedimos M y N que deben de ser inferiores al tamaño real de la matriz.

7. (*) Dadas dos matrices de 4×4 caracteres simularemos el juego de ‘hundir la flota’.
- Ambas matrices se rellenarán con caracteres ‘a’ (que simularán agua.)
 - Por teclado se introducirá 5 posiciones por matriz que rellenaremos con ‘b’ (barco)
 - Cada jugador realizará una jugada dando las coordenadas i, j de la matriz. Si la coordenada tiene ‘b’ pasará a ‘a’. El primero que hunda toda la flota gana y termina el juego.
 - Ampliar las funciones para que se presente un mensaje de ‘hundido’ solo si una posición marcada con ‘b’ como barco no tiene vecinos que tengan el valor ‘b’ (se ha hundido el barco completo).

Duración de la práctica: 2 sesiones

(*) Práctica obligatoria.

() Práctica obligatoria en la que hay que entregar el diagrama de flujo completo.**