

FUNDAMENTOS DE PROGRAMACIÓN

Tema 6. Arrays y matrices

1. Dado el siguiente segmento de programa:

```
void main()
{
    int n=0, i, j;
    int v1[25], v2[20];
    float m;
```

Indicar qué sentencias de la función **main()** son correctas (si fuesen incorrectas indicar el porqué):

- a) for (i=0; i<=20; i++)
 v1[i] = v2[i];
- b) m = v1[7] + v2[8];
- c) j=18; for (i=10; i<20; i++)
 v1[j] = v2[i]; j++;
- d) i=0; while (i<20) v1[i]=n+i;
- e) v1[25] = v2[20];

2. Dado el siguiente segmento de programa:

```
int media (int a, float x[15]); //prototipo de la función media
int n, m;
float v[15];
```

¿Cuáles de las siguientes sentencias, que incluyen una llamada a la función **media** son correctas?

- a) media (n, v) = m;
- b) m = media (n, v[15]);
- c) m = media (n, v);
- d) printf ("%f", media(n, v));
- e) m = media(m, v[15]) + m;

3. Escribe la declaración de una variable, **V**, de tipo vector de **N** enteros (N debe ser una constante) y escribe el código necesario para **dar la vuelta a los elementos del vector** (el primero pasa a ser el último, el segundo pasa a ser el penúltimo, etc). Considera dos casos diferentes (ambos deben emplear un esquema de **recorrido**):
- a. El vector resultante queda en una **variable distinta** a la original.
 - b. El vector resultante queda en la **misma variable** que el vector original.
4. Codificar una función que determine si un vector de componentes enteras es simétrico respecto al elemento central. La función devuelve uno (1) si el vector es simétrico y cero (0) en caso contrario. Por ejemplo, el vector 1 4 7 8 7 4 1, es simétrico.

5. Dado el siguiente programa:

```
#include <stdio.h>
/* declaracion de funcion Escribir y Rotar */

void Escribir (int N[10], int cant);
void Rotar(int N[10], int cant, int veces);

void main ()
{
    int i, Num[10];
    for (i= 0; i< 10; i++)
        Num[i] = i;
    Rotar (Num, 3, 5); //el 3 indica los numeros que se imprimen en cada línea
                      //el 5 indica el número de líneas que queremos imprimir
}
/* definicion de la funciones Escribir y Rotar */

void Escribir (int N[10], int cant)
{
    int i;
    for(i = 0 ; i < cant ; i++)
        printf("%d",v[i]);
    printf("\n");
}
```

Se pide escribir la función **Rotar** (NO recursiva) para que al ejecutar el programa la salida en pantalla sea, según el ejemplo:

Ejemplo:

Si la llamada es: Rotar(Numl, 3, 5);

La salida será:

012
201
120
012
201

Si la llamada es: Rotar(Num, 4, 7);

La salida será:

0123
3012
2301
1230
0123
3012
2301

Se deberá utilizar también la función **escribir** que cada vez que se le llame, escriba una línea de las que se pide.

1. Dadas las siguientes declaraciones:

```
void main()
{
    int sala[20][40]; //matriz de 20 filas y 40 columnas
                        //representa la sala de un cine
    int f, c;
```

Escribir el código necesario para:

- Poner todas las componentes a cero para indicar que la sala está vacía.
- Reservar de la tercera fila los asientos 10 y 11 poniendo a 1 dichos asientos.
- Reservar entera la quinta fila.
- Comprobar si quedan asientos libres en la decima fila.

Realizar una función que dada una matriz cuadrada de 5x5 escriba un mensaje en pantalla indicando si es simétrica o no. Una matriz es simétrica cuando para todo i, j valores de los índices $M[i][j] = M[j][i]$. No tratar más datos de los necesarios.

2. Una empresa tiene 20 tiendas, cada una de las cuales tiene 10 departamentos. Se dispone de una matriz de 20x10, que contiene las ventas de cada departamento en cada tienda. Escribir un programa que:

- Imprima el total de ventas de cada tienda con el siguiente formato:

TIENDA	TOTAL VENTAS
1	XXXX
2	XXXX
...

- Imprima el total de ventas de toda la empresa.
- Imprima el total de ventas de cada departamento, con el formato del apartado a.
- Imprima el departamento con menor número de ventas junto con la tienda a la que pertenece.

Esta función debe:

1. Introducir por entrada estándar la **fila** y la **columna** que se desea eliminar de la matriz A, comprobando que estos valores se encuentran en el rango de filas y columnas permitidas. Por ejemplo, en una matriz 3 x 4 no se puede eliminar la fila -1, ni la columna 5.
2. Construir una matriz auxiliar **B** de M - 1 filas y N - 1 columnas que contenga los elementos de la matriz original, excepto los elementos de la fila y la columna que se quiere eliminar.
3. Escribir en la salida estándar la matriz B resultante por filas.

Ejemplo:

fila y columna a eliminar: **2 3**

$$\mathbf{A} = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{array}$$
$$\mathbf{B} = \begin{array}{ccc} 1 & 2 & 4 \\ 9 & 10 & 12 \end{array}$$

Tema 5. Estructuras de Datos: Cadenas

1. Escribir las siguientes **funciones**:
 - a. Función **longitud**, que a partir de una cadena de caracteres nos devuelve su longitud.
 - b. Función **invertir**, que a partir de una cadena de caracteres devuelve otra cadena inversa a la original.
 - c. Función **iguales**, que a partir de dos cadenas devuelve un 1 si las cadenas son iguales y 0 en caso contrario.

NOTA: Suponer que las cadenas tienen un tamaño máximo de N=20 que incluye el final de cadena.

2. Utilizando las funciones anteriores, escribir un **programa** en C, que lea las palabras que hay en un fichero de texto **palabras.txt** y muestre en pantalla todas las palabras que son palíndromos (en el fichero *palabras* almacenamos en cada línea una palabra).

Una palabra es un palíndromo si se lee de igual forma de derecha a izquierda que de izquierda a derecha. Por ejemplo: *salas* es palíndromo.

3. Escribir un **programa** en C, que lea las palabras que hay en un fichero de texto **palabras2.txt** y muestre en pantalla un mensaje indicando cuántas palabras son iguales a la primera (en el fichero *palabras2* almacenamos en cada línea una palabra). Se pueden utilizar las funciones del ejercicio 1 que sean necesarias.

Tema 5. Estructuras de Datos: Arrays y Punteros

1. Escribir un programa que:
 - a. Cree un vector de enteros positivos que se leen desde el fichero estándar de entrada, no se conoce el número de elementos del fichero, se sabe que la secuencia termina con un número negativo y que el tamaño máximo es N.
 - b. Calcule la cantidad de números pares del vector y el número de impares.
 - c. Cree dos nuevos vectores uno de números pares y otro de impares. Libere el espacio no utilizado
 - d. Muestre por pantalla los dos nuevos vectores.