

Febrero 2005

NOTA IMPORTANTE.- Debes entrar en la cuenta que figura en tu examen (login y password). No puedes salir de esta cuenta durante el examen y solo puedes utilizar un editor, el compilador y el depurador. Lee con mucha atención antes de comenzar a trabajar.

TIEMPO: 2 HORAS 30 MINUTOS.

Recuerda que debes estructurar desde el principio tus ejercicios para tener un archivo principal (**llámalo examen.c**) en el que se encuentre sólo el main(), otro para las funciones (llámalo **funciones.c**), el correspondiente archivo de cabecera (**funciones.h**). El nombre del ejecutable deberá ser **examen.exe**. **Si no lo haces así restarás un punto de tu nota final. Recuerda que no debes utilizar variables globales en ningún caso. No tienes por qué hacer un menú en el programa principal. Puedes simplemente llamar a las funciones secuencialmente. Si al final tienes tiempo y haces un menú podrás sumar 1 punto más. El menú sólo te puntuará si tu nota es al menos de cuatro puntos.**

NO PODRÁS ABANDONAR EL AULA HASTA QUE FINALICE EL TIEMPO.

EJERCICIO 1 (3.5)

Se dispone de un vector de enteros de tamaño máximo 10:

- (1.5). Codifica una función de tipo void, llámala **LeeVector**, que lea desde teclado números enteros y los introduzca en el vector. La entrada de números finalizará cuando se lea un valor negativo, que no deberá considerarse, o cuando se hayan introducido 10 elementos. La función deberá devolver el tamaño final del vector (número de elementos en el vector).
- (2.0). Codifica una función de tipo void, llámala **Busqueda**, que reciba el vector de números enteros, su tamaño y dos parámetros de tipo entero pasados por referencia. La función deberá determinar, de todos los valores del vector, cual es el número par más pequeño y el impar más grande y devolverlos usando los parámetros pasados por referencia. El programa principal imprimirá estos dos valores.

Ejemplo de prueba:

Elementos del vector: 7, 26, 3, 2, 15, 18, 42, 111

Debe salir: Número par: 2 Número impar: 111

EJERCICIO 2 (3.0)

Un palíndromo es una cadena de caracteres que se lee igual de derecha a izquierda que de izquierda a derecha, independientemente de los espacios en blanco que contenga. Por ejemplo:

*“yo hago yoga hoy”
“dabale arroz a la zorra el abad”*

Escribe una función **palindromo** en C para averiguar si una cadena de caracteres es un palíndromo. La función acepta como parámetro una cadena de caracteres y devuelve 0 si la cadena no es palíndromo y 1 si lo es. Suponer un tamaño máximo de cadena de 100 caracteres (incluido el carácter de fin de cadena).

EJERCICIO 3 (3.5)

Se tiene una matriz real (float), con un máximo de 10 filas y 10 columnas. Codifica los siguientes módulos:

- (0.75). Una función, llámala **LeeMatriz**, que lea de teclado los elementos de la matriz. La función deberá preguntar al usuario el número de filas y columnas que tendrá la matriz y posteriormente devolverlos al programa principal para utilizarlos posteriormente.
- (0.75). Una función, llámala **ImprimeMatriz**, que reciba las dimensiones de la matriz (dos enteros) y la matriz, e imprima por pantalla los elementos que la componen.
- (2.0). Una función, llámala **Inversión**, que reciba la matriz y las dimensiones de la misma (dos enteros), y realice una inversión de la matriz por columnas. La función deberá intercambiar la primera columna de la matriz por la última, la segunda por la penúltima y así sucesivamente hasta llegar al centro de la matriz. La inversión se hará sobre la misma matriz.

Ejemplos de prueba:

- Matrices con un número de filas impar

```
1  2  3          3  2  1
4  5  6          6  5  4
7  8  9 => salida => 9  8  7
10 11 12        12 11 10
```

- Matrices con un número de filas par

```
5  6  7  8          8  7  6  5
3  4  2  1 => salida => 1  2  4  3
```