

—NO SE ADMITIRÁ NINGÚN EXAMEN ESCRITO A LÁPIZ—

—SE PUEDE ESCRIBIR POR AMBAS CARAS DE CADA HOJA DE EXAMEN—

---

Los problemas 4º y 5º suman un total de 4 puntos y han de ser realizados por todos los alumnos. El resto del examen, compuesto por el test y los problemas 1º, 2º y 3º, únicamente es para aquellos alumnos que no superaron la evaluación continua, es decir, que han obtenido una nota inferior a 3 sobre 6 al sumar la nota sobre el trabajo del primer tema y la de la prueba de los boletines. Cada una de las preguntas de TEST vale 0.25 puntos. Los fallos no restan.

---

**TEST (1.5 puntos):** (Se contesta en hojas aparte)

1. Indicad cuál de las siguientes afirmaciones es cierta:
  - a) El *byte* es la unidad mínima de información en un ordenador.
  - b) Con  $n$  *bits* podemos codificar  $2^n$  valores diferentes.
  - c) Actualmente todos los ordenadores codifican los números mediante la tabla ASCII.
  - d) 1024 *Kbytes* equivalen a  $2^{10}$  *bytes*.
2. La memoria RAM es un dispositivo que:
  - a) Almacena las instrucciones de un programa, donde se indican las operaciones a realizar, pero no los datos, puesto que éstos se almacenan sólo en disco.
  - b) Organiza la información mediante palabras y un dato puede ocupar una o más palabras.
  - c) Almacena la información de manera permanente.
  - d) Para acceder a una palabra es necesario pasar antes por todas las anteriores.
3. Indicad cuál de las siguientes afirmaciones es cierta:
  - a) Todo periférico necesita un controlador o *driver* que le permita comunicarse con el usuario.
  - b) Todo aquello que en un ordenador está dentro del sistema central se agrupa bajo el término de periférico.
  - c) Los periféricos suplen la falta de velocidad de la memoria central.
  - d) Los dispositivos de almacenamiento secundario también se clasifican como periféricos.
4. Indicad cuál de las siguientes afirmaciones es cierta:
  - a) Los buses suponen el medio de comunicación entre los diferentes componentes de un computador.
  - b) La falta de memoria RAM a la hora de ejecutar un programa se suple habitualmente aumentando la memoria caché del procesador.
  - c) Un sistema operativo se encarga de controlar la ejecución de los procesos de usuario pero no de gestionar los recursos *hardware*.
  - d) La unidad aritmético-lógica extrae de la memoria central las instrucciones a ejecutar, y la unidad central realiza las operaciones apropiadas con los datos.
5. ¿Cuál de las siguientes características **no** pertenece a los discos duros?
  - a) Guarda la información dibujando símbolos sobre las pistas.
  - b) Cada pista se divide en sectores.
  - c) Guarda la información sobre un medio magnético organizada en pistas concéntricas.
  - d) Es un dispositivo de almacenamiento secundario.

6. ¿Cuál de los siguientes componentes no forma parte de la Unidad Central de Proceso (CPU) ?
- a) Memoria Central y Buses de Interconexión.
  - b) Reloj.
  - c) Unidad Aritmético-Lógica (UAL) y Registros.
  - d) Unidad de Control (UC).

**PROBLEMAS:** (Se contestan en hojas aparte)

1. (1.5 puntos)

Desarrollad una función que devuelva el tamaño de un fichero (número de caracteres). Se le pasará como parámetro una cadena de caracteres con el nombre (la ruta completa) del fichero. La función deberá leer todos los caracteres del fichero, retornando el número de caracteres leídos.

Nombrando a la función solicitada `longitud_fichero()`, en el siguiente programa se ilustra un ejemplo de su uso.

```
#include <stdio.h>

int main()
{
    int    tamaño;

    tamaño = longitud_fichero( "ejemplo.txt" );
    printf( "La longitud del fichero es de %d bytes.\n", tamaño );
    return 0;
}
```

NOTA: Para detectar el final de fichero se puede utilizar la función `feof()`, que devuelve 1 cuando se ha alcanzado el final del fichero y 0 mientras no.

2. (1.5 puntos)

Ofreced una implementación alternativa de la siguiente función sin hacer uso de la sentencia `switch`.

```
void f( int a )
{
    switch( a ) {
        case 1 : printf( "Uno\n"); break;
        case 2 : printf( "Dos\n"); /* OJO que aquí no hay break */
        case 3 : printf( "Tres\n" ); break;
        default : printf( "Otros\n" );
    }
}
```

3. (1.5 puntos)

Implementad un programa en lenguaje **C** que pida una cadena de caracteres al usuario y la vuelva a imprimir por pantalla toda con mayúsculas. Para convertir una letra minúscula a mayúscula se puede utilizar la función `toupper()`, que toma un carácter, y si se trata de una letra minúscula devuelve su correspondiente mayúscula, en cualquier otro caso devuelve el mismo carácter. Para el uso correcto de esta función es necesario incluir el fichero `ctype.h`.

4. (3 puntos)

La organización del *rally* París-Dakar nos pide que realicemos un programa para llevar el control de los **P** participantes de una sola categoría, por ejemplo coches. Considerando una única categoría el programa a realizar es más sencillo.

El programa a implementar (en lenguaje **C**) ha de ser capaz de almacenar los tiempos en minutos que cada uno de los **P** participantes ha empleado en completar cada una de las **E** etapas. **P** y **E** serán dos constantes enteras definidas por el programador, indicando el número de participantes y de etapas respectivamente. Con los tiempos almacenados el programa nos indicará el ganador de cada etapa y el ganador del *rally*.

Para simplificar, además de considerar una única categoría no será necesario guardar el nombre del participante ni del equipo, simplemente cada participante será identificado por un número entero del **1** al **P**.

El programa deberá mostrar un menú con diferentes opciones para que el usuario pueda realizar las siguientes acciones:

1. Introducir por teclado los tiempos. Esta función puede hacerse de dos maneras: a) Para ser invocada cada vez que un participante llega a meta, de manera que habrá que introducir etapa, participante y minutos. b) Que pida los datos de todos los participantes y todas la etapas de golpe.
2. Listar el ganador de cada etapa (el que menos tiempo ha necesitado).
3. Mostrar el ganador del *rally* (el que menos tiempo ha necesitado considerando todas las etapas).
4. Salir del programa.

Excepto la opción de salir del programa, las restantes opciones serán implementadas mediante funciones. De no ser así, el programa, aunque esté bien hecho, puntuará 1 punto, y no los 3 puntos. Además, la implementación del programa no podrá utilizar variables globales. La o las variables que se utilicen para almacenar los datos se pasarán de una función a otra mediante el paso de argumentos pertinente, es decir, como se ha hecho en clase.

5. (1 punto)

Escribid una función que, recibiendo un vector de enteros positivos de tamaño **N**, donde **N** será una constante entera definida por el programador, escriba en pantalla los valores del vector, uno por línea, junto a la palabra **Par** o **Impar** según corresponda.

Los números han de estar bien alineados como se indica en el siguiente ejemplo:

```
28 Par
29 Impar
120 Par
 1 Impar
 99 Impar
1231 Impar
 17 Impar
 44 Par
```

Nota: únicamente se pide la función que recibe como argumento (o parámetro) un vector de enteros e imprime por pantalla lo solicitado. No se pide un programa completo.