

| | | | | | |
|-------------|--|---------------------|--|--------------------|--------|
| Aula | | Fila | | Col. | |
| Nombre | | 1 ^{er} Ap. | | 2 ^o Ap. | |
| | | | | | Firma: |

NOTAS DE OBLIGATORIA LECTURA

1. No se admitirá ningún examen escrito a lápiz
2. Sé ecológico: escribe por ambas caras de cada hoja de examen
3. Para la realización de este examen es obligatoria la presentación de identificación oficial actualizada con fotografía: carnet de estudiante, dni, pasaporte
4. El examen de tipo test se entrega junto con el resto del examen. Todas las preguntas de desarrollo se contestan en hoja aparte y en cualquier orden. Pueden contestarse varias preguntas en la misma hoja

TEST (1.5 puntos). Los aciertos cuentan como 0.3 puntos. 3 fallos descuentan 1 acierto.

1. Indicar cuál de las siguientes funciones es una tarea del sistema operativo
 - a. La reproducción de ficheros de audio
 - b. La gestión de memoria
 - c. El navegador de internet
 - d. La gestión del sistema de citas de la agenda electrónica
2. Indicar cuál de las siguientes afirmaciones es VERDADERA
 - a. El esquema general de funcionamiento de un programa consiste siempre en dos únicas partes: entrada de datos y proceso de los resultados
 - b. Los programadores escriben los programas en código binario
 - c. Los lenguajes de alto nivel son aquellos en los que las instrucciones se expresan según las particularidades de cada CPU
 - d. Un lenguaje de programación es un conjunto de reglas, símbolos y palabras especiales utilizados para construir programas
3. Indicar la opción correcta:
 - a. Los ordenadores procesan la información en formato analógico
 - b. Con n bits se pueden representar n*2 valores diferentes
 - c. 1 Kbyte = 8192 bits
 - d. El sistema más extendido para el almacenamiento de caracteres en un ordenador es el ASCII, donde se usan 6 bits para cada carácter
4. Dado el siguiente bucle, `for(t=0.0; t < 100.0; t+=0.5) { }`, indicar cual de los siguientes deja en la variable 't' el mismo resultado que el anterior tras finalizar su ejecución
 - a. `for(t=-0.0; t <= 100.0; t= t + 0.5) { }`
 - b. `t=-0.0; while(t <= 100.0) { t= t + 0.5; }`
 - c. `t=-11.5; do { t= t + 0.5; } while (t < 100.0);`
 - d. `for(t=100.0; t < 100.5; t= t + 0.5) { }`
5. ¿Puede un programa escrito en lenguaje C ejecutarse sin contener la función main() ?
 - a. Sólo cuando se ejecuta dentro del entorno Linux. En Windows, no.
 - b. No, nunca
 - c. Sí, pero sólo cuando contiene la cláusula #include
 - d. Sí, pero sólo cuando contiene la cláusula #define

ENUNCIADOS



1.- (4 puntos) La organización de los próximos Juegos del Mediterráneo de Almería nos pide que desarrollemos un programa en lenguaje C que gestione las carreras de los deportistas durante el desarrollo de la competición.

El programa debe considerar un total de D deportistas (identificados por un número del 1 al D, ambos inclusive) que compiten en un total de C competiciones (identificadas por un número del 1 al C, ambos inclusive), donde D y C deben definirse como constantes.

El programa mostrará por pantalla un menú compuesto por las diferentes opciones:

1. Mostrar el ganador de una carrera concreta (el que menos tiempo ha invertido). (1.0 puntos)
2. Mostrar por pantalla el tiempo medio que un participante ha empleado en recorrer las diferentes carreras. (1.0 puntos)
3. Mostrar por pantalla el ganador de la competición (el que menos tiempo ha invertido en recorrer todas las carreras). (1.0 puntos)
4. Salir del programa.

En el momento de comenzar, y antes de mostrar el menú por pantalla, el programa leerá los tiempos invertidos por todos los deportistas en todas las competiciones desde disco. La función encargada de leer los tiempos desde disco ya está implementada, por lo que no hay que desarrollarla, sino sólo utilizarla. Su perfil es el siguiente:

```
void LeerTiempos (float Tiempos[C][D]);
```

Cada una de las opciones, incluida la presentación del menú en pantalla y la lectura de la opción seleccionada, debe estar implementada por una función independiente. La única opción que no se implementará mediante una función es la opción *salir*. El programa no podrá emplear variables globales. (Menú y programa principal: 1.0 puntos).

Almacenar los tiempos internamente en segundos. Obviamente, los tiempos nunca pueden ser negativos.

2.- (2.5 puntos) Escribir un programa en lenguaje C que solicite al usuario por el teclado dos datos (edad y sexo) para cada una de las personas de un colectivo.

- La edad será un número entero positivo o cero.
- El sexo será un carácter con el valor 'H' para hombre o 'M' para mujer.

Si el usuario introduce, por error, un carácter distinto para el sexo, el programa deberá volver a solicitar el dato hasta que introduzca un carácter válido: 'H' o 'M'. La introducción de datos finalizará cuando el usuario introduzca la edad cero.

Tras la entrada de datos, se deberá mostrar por pantalla la siguiente información:

- el número total de personas,
- el número de hombres, cuántos de ellos son menores de edad (<18 años) y la media de edad para todos los hombres,
- el número de mujeres, cuántas de ellas tienen una edad mayor o igual a 50 años y la media de edad para todas las mujeres.

A continuación se muestra un ejemplo de la ejecución del programa:

Edad: 15
Sexo: P

Incorrecto, pruebe de nuevo

Sexo: H

Edad: 55

Sexo: M

Edad: 30

Sexo: M

Edad: 0

En este grupo hay 3 personas, de las cuales:

-Hombres: 1 , menores de edad: 1 , media: 15 años

-Mujeres: 2 , mayores de 50: 1 , media: 42.5 años

3.- (2 puntos) Se dispone de un fichero de texto con información acerca de la conversión entre distintas unidades de medida denominado "ConvUni.txt". El fichero contiene, en su primera línea, un número entero que indica cuantas conversiones de unidades de medida se incluyen a continuación. A partir de la segunda línea, aparece en cada línea el nombre de una unidad de medida, seguido de un espacio en blanco, seguido del factor de conversión en formato numérico con decimales, seguido de otro espacio en blanco y finalmente el nombre de otra unidad de medida. Todas las palabras están escritas en minúsculas.

Un ejemplo de un posible contenido válido del fichero "ConvUni.txt":

3

metro 39.37 pulgada

kilometro 39370 pulgada

pulgada 2.54 centímetro

Se pide que se implemente un programa en lenguaje C que solicite al usuario que introduzca por teclado dos cadenas de texto que identifiquen a la unidad a convertir y la unidad convertida (ambas en minúsculas). Una vez obtenidos dichos valores, el programa deberá buscar si existe la conversión entre ambas unidades en el fichero "ConvUni.txt". En caso de que exista dicha conversión, deberá mostrar por pantalla un mensaje indicando la unidad a convertir, el factor de conversión y la unidad convertida. Si no existe, se indicará que dicha conversión no existe. El programa acabará cuando se introduzca la palabra "fin" como unidad a convertir.

Ejemplo de ejecución del programa

Si se le pasa "pulgada" y a continuación "metro" (la conversión de pulgada a metro no existe en el fichero "ConvUni.txt") visualizará por pantalla el texto "Conversión de pulgada a metro no disponible".

Si se le pasa "pulgada" y a continuación "centímetro" (la conversión de pulgada a centímetro si que existe en el fichero "ConvUni.txt") visualizará por pantalla el texto "1 pulgada = 2.54 centímetro".

NOTA: Para la comparación de cadenas de texto, se aconseja utilizar la función denominada `int strcmp (char*, char*)`; incluida en la biblioteca `string.h`.

Esta función permite comparar dos cadenas de caracteres (que se pasan como parámetros) y devuelve el valor 0 cuando son iguales y cualquier otro valor cuando son distintas.