

1. La asociación de montañistas evaluó las posibilidades de supervivencia en las excursiones y descubrió que hay ciertas reglas que siempre se dan:

- Cada excursionista come 1.5 kilos de comida al día.
- Si les alcanza la comida para la duración de TODA la excursión todos los excursionistas se salvan.
- Si la comida no alcanza para alimentarlos a todos durante la duración de la excursión, el número de excursionistas que se salvan depende de los factores de la siguiente tabla:

Estado del tiempo	Días que restan de la excursión después de que se les acaba la comida	Número de excursionistas que se salvan
No importa	0 a 3	Todos
Bueno	Más de 3 hasta 5	Todos
Malo	Más de 3 hasta 5	La mitad
No importa	Más de 5	Ninguno

Realizad un programa de ordenador en lenguaje **C** que pida la cantidad de comida en kilos, el número de excursionistas, duración de TODA la excursión y el estado del tiempo (1 = Buen tiempo, 0=Mal Tiempo), y diga cuántos excursionistas se salvan.

2. Realizad un programa de ordenador en lenguaje **C** que pida al usuario la hora actual (horas y minutos) y la duración en minutos de la cocción de un guiso, entonces que muestre por pantalla la hora en que el guiso estará listo.

Las horas van de 0 a 23 para no tener que pedir al usuario si es mañana o tarde.

3. Realizad un programa de ordenador en lenguaje **C** que pida por teclado un valor entero X y que posteriormente escriba por pantalla todas las horas del día, en el formato hh:mm (ejemplo: 12:15) tales que la suma de las cifras (por ejemplo, para las 12:15 la suma es $1+2+1+5=9$) sea igual a X.
4. Realizad un programa de ordenador en lenguaje **C** que lea por teclado un numero N entre 1 y 20 y escriba un cuadrado de lado N formado por: asteriscos (*) en la parte inferior de la diagonal principal, guiones (-) en la diagonal principal y signos mas (+) por encima de la diagonal principal. El programa debe dar un mensaje de error cuando el numero N no está en los límites indicados y solicitar nuevamente el valor N al usuario.

Ejemplo para n=4:

```
-+++  
*-++  
**-+  
***-
```

5. En una escuela hay 3 pisos con 4 aulas en cada piso. Todos los días se guarda relación del número de alumnos que han asistido a cada aula. Se pretende escribir un programa en lenguaje **C** que, introduciéndole estos datos (el número de alumnos que han asistido a cada aula), muestre el número de alumnos que han asistido en total en cada piso, así como el número total de alumnos que han asistido entre todas las aulas de todos los pisos.

6. La sección de moda joven de unos grandes almacenes desea informatizar el pago de comisiones a cada uno de sus 10 vendedores. Al comienzo de cada día, el programa se pone en marcha, con un vector que almacena el dinero vendido por cada vendedor e inicializado a cero. Conforme va transcurriendo el día, y tras cada venta, cada vendedor se identifica en el ordenador e introduce la cantidad en Euros correspondiente, de manera que, automáticamente, el programa incrementará el dinero recaudado por dicho vendedor. Al finalizar el día, lo cual se indica al introducir el vendedor -1, el programa deberá mostrar el dinero recaudado por cada vendedor, junto con su comisión correspondiente, la cual se calcula siguiendo la siguiente tabla:

Recaudación diaria (en Euros)	Comisión
[0, 180[—
[180, 320[2 %
[320, 600[2,5 %
[600, 900[3,5 %
≥ 900	5 %

7. El propietario de un videoclub nos pide que desarrollemos un programa en lenguaje **C** que gestione el alquiler de las películas de su negocio. Se supone que cada película se identifica mediante un número. El programa debe constar de un menú capaz de:

- Inicializar la cantidad de copias existentes de cada película.
- Alquilar una película. Si existen copias disponibles en el videoclub, el número de copias se reducirá en una unidad. En caso contrario se deberá mostrar un mensaje indicando que no se puede alquilar la película.
- Devolver una película. El número de copias disponibles de una película concreta se incrementa en una unidad.
- Consultar el número de copias disponibles. Tras pedir el número de película, se muestra el número de copias disponibles en la sala.

8. Se tienen dos vectores de 500 elementos cada uno. Un vector contiene las alturas y el otro los pesos de 500 niños/as. Los elementos pares de los vectores corresponden a la altura/peso de niños, y los elementos impares a los datos de las niñas. Los primeros 100 elementos de los vectores registran datos de niños/as de 10 años, los siguientes 100 elementos de niños/as de 11 años, etc., y los últimos 100 elementos corresponden a datos de niños/as de 14 años. Se pide realizar un programa en lenguaje **C** que, dada una edad y un sexo, devuelva la altura y peso medio.

9. Escribid un programa en lenguaje **C** para comprobar el premio que le ha correspondido a un usuario que juega al cupón de la ONCE.

Para ello, el programa debe leer un fichero que contiene en cada fila el número agraciado y el premio que le ha correspondido. Por ejemplo: 12345 6000, es decir, al número 12345 le han correspondido 6000 euros.

Esta información se debe almacenar en una matriz con dos columnas y tantas filas como filas tenga el fichero.

A continuación, el usuario tecleará tantos números como cupones ha comprado y el programa indicará si ese número ha sido premiado y en ese caso, cuántos euros le han correspondido. El usuario introducirá el número -1 para indicar que ya no tiene más cupones que introducir.

El programa también mostrará al final la suma total de los premios que ha obtenido el usuario.

10. Se requiere un programa en lenguaje **C** que dibuje en pantalla un tablero de ajedrez mediante caracteres ASCII. Las fichas se escribirán de la siguiente forma:

P	Peón	A	Alfil	C	Caballo	T	Torre	D	Dama	R	Rey
----------	------	----------	-------	----------	---------	----------	-------	----------	------	----------	-----

La inicial se mostrará en mayúsculas para el bando blanco y en minúsculas para el bando negro.

Las casillas que no contengan nada tendrán dos espacios en blanco. Todas las casillas se separan entre si por barras horizontales o verticales. El tablero se numera de 1 a 8 las filas y de a a h las columnas.

El programa presentará un menú por pantalla que será:

1. Nueva partida.
2. Realizar movimiento.
3. Salir.

La opción realizar movimiento consiste en solicitar las usuario la casilla inicial y final. El movimiento tampoco se permite si en la posición inicial no hay ficha. Si en la posición destino existe una ficha del bando contrario, la elimina del juego. Posteriormente se mostrará el tablero actualizado.

Después de cada una de las opciones se mostrará el menú nuevamente.