EJERCICIOS DE PROGRAMACION MODULAR

Problema 1: Dibujando cuadrados

Realiza un programa en C que dibuje por pantalla un cuadrado. El programa debe tener además de la función main0 los siguientes módulos:

- leePar(): lee un carácter c1, un carácter c2 que debe ser ´R´ o ´V´ y un entero n entre 4 y 20.
- dibuja(): recibe como argumento los parámetros leídos en el módulo anterior y visualiza un cuadrado de lado n utilizando el carácter c1. Si el carácter c2 es ´R´ el cuadrado debe estar relleno, en caso de que sea ´V´ sólo debe dibujar el contorno. Este módulo debe devolver el número de caracteres dibujado y el tamaño del espacio interior (expresado como número de espacios en blanco del interior de la figura).

El main() debe mostrar en pantalla el número de caracteres dibujado y el tamaño del espacio interior.

Ejemplos de ejecución:

```
Introduce un carácter: *
Introduce otro carácter (R/V): R
Introduce un número (entre 4 y 20): 6
******
******
******
******
******
******
El número de caracteres dibujado es 36
El tamaño del espacio interior  0
```

```
Introduce un carácter: +
Introduce otro carácter (R/V): V
Introduce un número (entre 4 y 20): 3
3 no es válido
Introduce un número (entre 4 y 20): 6
++++++
+    +
+    +
+    +
+    +
++++++
El número de caracteres dibujado es 20
El tamaño del espacio interior  16
```

Problema 2: El blackJack

Implementa una versión simplificada del juego de cartas BlackJack, también llamado 21. En este juego se van repartiendo cartas de una en una y el objetivo del juego es llegar lo más cerca posible de 21 puntos sin pasarse. Habitualmente participa la banca y uno o más jugadores. La mecánica del juego será la siguiente:

- Para simplificar, en nuestra versión del juego solo participa un jugador y la banca.
- La banca no va a ir obteniendo cartas como el jugador, simplemente se le asignará una puntuación aleatoria entre 1 y 21.
- Al jugador se le irán “repartiendo” cartas de 1 en 1. Para esto, se generará un valor al azar entre 1 y 12. (Para simplificar, no tendremos en cuenta el palo de la carta).
- Tras “repartirle” carta al jugador se le dirá qué carta ha salido, cuántos puntos lleva, y, si no se ha pasado de 21, se le preguntará si quiere otra carta (tendrá

que introducir una ‘s’ o una ‘n’). Si no introduce ninguna de las dos, se le volverá a preguntar.
- Cuando el jugador diga que no quiere más cartas, se mostrará el resultado final. Ganará el jugador si tiene más puntos que la banca y no se ha pasado de 21 (fijaos en que la banca no puede pasarse tal y como funciona nuestra versión del juego)

El programa deberá tener al menos los siguientes módulos:

- reparteCarta: debe devolver el número de la carta “repartida” (o sea, generada al azar). Tendréis que generar un entero al azar entre 1 y 12. No es misión de este módulo imprimir en pantalla qué carta ha salido, esto es trabajo del main o de otro módulo.
- calculaTotal: recibe como parámetro los puntos actuales del jugador y el número de la carta que ha salido. Devuelve el nuevo total, sumando los puntos actuales más el valor de la nueva carta teniendo en cuenta que:
    - El as (1) vale 11 puntos.
    - Las figuras (cartas mayores que 9) valen 10.
    - El resto de cartas tienen su valor habitual.
    - No es misión de este módulo comprobar si nos hemos pasado o no de 21, simplemente debe calcular el resultado (sea el que sea).

Ejemplos de ejecución:

```
Te ha salido un 2
Por ahora tienes 2. ¿Quieres carta? (s/n) s
Te ha salido un 9
Por ahora tienes 11. ¿Quieres carta? (s/n) k
¿Quieres carta? (s/n) s
Te ha salido un 7
Por ahora tienes 18. ¿Quieres carta? (s/n) n
La banca tenía 6
Has ganado!!!
-----------------------------------------------
Te ha salido un 5
Por ahora tienes 5. ¿Quieres carta? (s/n) s
Te ha salido un 7
Por ahora tienes 12. ¿Quieres carta? (s/n) s
Te ha salido un 11
Te has pasado
```

```
La banca tenía 20

Ha ganado la banca
```

Problema 3: El e-Tutor 2.0

En el colegio han quedado encantados con el programa que hicimos para ayudar a los niños a aprender a multiplicar, así que nos han pedido que lo mejoremos para ayudarles a practicar y aprender con todas las operaciones. En este caso, el programa debe generar 7 operaciones consecutivas que el estudiante debe ir respondiendo correctamente para ir avanzando. Los números aleatorios que genera el ordenador deben estar entre 1 y 9 y las operaciones aritméticas a considerar son la suma, la resta, la multiplicación y la división entera. El resultado de una operación (si es correcta) se convierte en el primer operando de la siguiente operación. La instrucción termina cuando el estudiante ha acertado 7 operaciones. Cuando falla en una operación, se vuelve a generar otro operador y otro valor para el segundo operando. El diseño del programa debe ser modular.

Ejemplo de ejecución:

```
3-3= 0
¡¡HAS ACERTADO!!
0+1= 1
¡¡HAS ACERTADO!!
1*6= 7
Ups. Vuelve a intentarlo
7/6= 1
¡¡HAS ACERTADO!!
1/1= 1
¡¡HAS ACERTADO!!
1-2= -1
¡¡HAS ACERTADO!!
-1*1= -1
¡¡HAS ACERTADO!!
-1*4= -4
¡¡HAS ACERTADO!!
```