

## Ejercicio 1.-Contador de caracteres

Se pretende generar un programa que cuente cuantas veces se repiten ciertos caracteres hasta un máximo de 10 caracteres. Al invocar el programa se indicarán los caracteres que se quieren contar separados por espacios. Una vez arrancado el programa se ofrecerá un menú desde el que se podrá introducir un nuevo texto (de máximo 40 caracteres), mostrar el resultado o salir. Cada vez que se introduce un nuevo texto los contadores no se inicializan, sino que siguen contando desde el último valor.

```
$ ./contar.out a b c
-----
Para introducir una nueva frase pulse 1
Para mostrar el resultado pulse 2
Para salir pulse 0
-----
1
Introduce el texto:
Siempre le pregunto a la razon
-----
Para introducir una nueva frase pulse 1
Para mostrar el resultado pulse 2
Para salir pulse 0
-----
1
Introduce el texto:
Y me contesta que tal vez
-----
Para introducir una nueva frase pulse 1
Para mostrar el resultado pulse 2
Para salir pulse 0
-----
2
a      5
b      0
c      1
-----
Para introducir una nueva frase pulse 1
Para mostrar el resultado pulse 2
Para salir pulse 0
-----
0
Este programa ha terminado...
$
```



## -Ejercicio 2.-Batalla de gladiadores

Simular una batalla entre dos gladiadores, uno será el usuario y el otro la máquina.

Cada gladiador tiene un nombre (hasta 10 caracteres), un tipo de arma, una fuerza y una salud.

Los ataques dependerán de la fuerza del gladiador y del tipo de arma, restarán salud a su contrincante. Deberá haber dos tipos de arma:

- Espada: el ataque será la fuerza del gladiador.
- Lanza: en este caso será el doble de la fuerza del gladiador.

El programa deberá permitir al usuario ingresar los nombres de los dos gladiadores y datos de juego desde la terminal. Al ejecutar el programa se le pasarán los datos de dos gladiadores (usuario y cpu en ese orden). Deberá pasarle los siguientes parámetros (Espartaco será el usuario, Marcus la máquina):

```
./gladiadores Espartaco,20,espada Marcus,10,lanza
```

Al inicio del programa, ambos jugadores tienen una salud de 100 unidades. Durante la batalla, el usuario tendrá la oportunidad de atacar o defenderse en cada turno mediante un menú. En el caso de **la máquina atacará con una probabilidad del 50% aleatorio**.

La batalla se desarrollará por turnos **hasta que uno de los gladiadores quede sin salud**.

Al finalizar la batalla, el programa deberá mostrar el resultado y mostrar los datos del ganador de la batalla.

Ejemplo de ejecución:

```
$ ./gladiadores.exe Espartaco,20,espada Marcus,10,lanza
Turno de Espartaco:
1. Atacar
2. Defender
Selecciona una opcion: 1
Atacaste a Marcus y le hiciste 20 de danio!
Turno de Marcus:
Marcus te ataca y te hace 20 de danio!
Turno de Espartaco:
1. Atacar
2. Defender
Selecciona una opcion: 2
Te has defendido.
```

```
Has ganado la batalla!
Nombre: Espartaco
Tipo de arma: Espada
Fuerza: 20
Salud: 20
```



### Ejercicio 3.-Torneo de pádel

Se va a organizar un torneo de pádel con 4 parejas fijas. Todas las parejas jugarán entre sí en 3 jornadas. El programa debe guardar los datos de las parejas y los resultados de los partidos y obtener una clasificación.

Los nombres (sólo el primer nombre o apodo sin espacios, no deben estar repetidos, tamaño máximo de 10 caracteres) de los jugadores se deberán introducir al invocar el programa y las parejas se formarán en el orden en el que se introduzcan los jugadores. Los dos primeros nombres formarán una pareja, el tercero y el cuarto otra y así sucesivamente.

Por menú se podrán solicitar las siguientes opciones:

- Introducir un resultado de una jornada. Se deberá guardar el número de jornada actual Para cada resultado se mostrarán los participantes de ese partido y el usuario deberá introducir el resultado separado por espacios. Los partidos son al mejor de 3 sets. Eso quiere decir que los posibles resultados son:

2      0

2      1

- Mostrar los resultados de todos los partidos. Se mostrarán los resultados de los partidos jugados hasta la fecha.
- Obtener la pareja ganadora. Esta opción únicamente se podrá ejecutar cuando se hayan introducido las tres jornadas. Si hay varios jugadores con el mismo número de partidos ganados el empate se deberá de resolver con número de sets a favor menos número de sets en contra. Si aun así sigue habiendo empate se mostrarán todas las parejas empatadas.

Se deberá utilizar tres tipos de estructuras, para guardar parejas, partidos y jornadas. Un ejemplo de ejecución con la introducción de nombres, menús, etc... sería la siguiente:

```
$ ./torneo.out juan pedro marta luis silvia rafa dani carlos
1-Introducir Resultados, 2-Mostrar resultados, 3-Mostrar ganador, 0-Salir
1
Resultado de la jornada 1
juan-pedro -- marta-luis: 1 2
silvia-rafa -- dani-carlos: 0 2
1-Introducir Resultados, 2-Mostrar resultados, 3-Mostrar ganador, 0-Salir
1
Resultado de la jornada 2
juan-pedro -- silvia-rafa: 2 1
dani-carlos -- marta-luis: 1 2
1-Introducir Resultados, 2-Mostrar resultados, 3-Mostrar ganador, 0-Salir
1
Resultado de la jornada 3
juan-pedro -- dani-carlos: 2 0
marta-luis -- silvia-rafa: 0 2
1-Introducir Resultados, 2-Mostrar resultados, 3-Mostrar ganador, 0-Salir
2
juan-pedro--marta-luis:1 2
silvia-rafa--dani-carlos:0 2
juan-pedro--silvia-rafa:2 1
dani-carlos--marta-luis:1 2
juan-pedro--dani-carlos:2 0
marta-luis--silvia-rafa:0 2
1-Introducir Resultados, 2-Mostrar resultados, 3-Mostrar ganador, 0-Salir
3
*****
juan - pedro
*****
1-Introducir Resultados, 2-Mostrar resultados, 3-Mostrar ganador, 0-Salir
0
$
```

#### Ejercicio 4.-Tienda de mesas

Una tienda que vende mesas bajo demanda necesita realizar una gestión sobre las mesas de las que dispone. Se quiere realizar pedidos usando un muestrario de mesas, que son las introducidas hasta ese momento. Una mesa almacena la siguiente información:

- Un tamaño “ancho/alto/fondo” dado en cm.
- Cada mesa puede tener un **tipo de material distinto**. Los materiales disponibles son
  - o Plástico
  - o Madera
  - o Cristal
  - o Metal
- Precio

Se pide realizar un programa que realice la siguiente funcionalidad:

- Al inicio del programa, se mostrará un menú con las siguientes opciones:
  - o 1- Introducir características de una nueva mesa
  - o 2- Mostrar las mesas disponibles
  - o 3- Calcular precio pedido
  - o 4- Salir
- Si se selecciona la opción 1, el programa permitirá introducir los datos de un modelo de mesa. Se deberá guardar en una lista para poder acceder en las siguientes opciones del programa, hasta un máximo de 100 modelos de mesas. No se comprueba si dos modelos son iguales.
- Si se selecciona la opción 2, se mostrarán los datos de mesas introducidas hasta ese momento.
- Si se selecciona la opción 3, el programa mostrará el número de mesas disponibles, y pedirá qué mesas se quiere comprar. El usuario **introducirá los índices** de las mesas elegidas dentro de las disponibles hasta ese momento, separados por comas, y con un máximo de 5 mesas. En ese momento se sumarán los precios de las mesas seleccionadas y mostrará el precio final al usuario.
- Si se selecciona la opción 4, el programa terminará.

Ejemplo de ejecución:

- Ejemplo introduciendo datos de mesas

```

1- Introducir características de una nueva mesa
2- Mostrar las mesas disponibles
3- Calcular precio pedido
4- Salir
1
Introduzca alto:
12
Introduzca ancho:
12
Introduzca fondo:
12
Introduzca material:
0-plastico
1-cristal
2-madera
3-metal
1
Introduzca precio:
12

```

- Ejemplo mostrando la lista de mesas:

```

1- Introducir características de una nueva mesa
2- Mostrar las mesas disponibles
3- Calcular precio pedido
4- Salir
2
=====
Mesa: 0
Alto:12
Ancho:12
Fondo:12
Material:cristal
Precio:12.000000
=====
Mesa: 1
Alto:13
Ancho:13
Fondo:13
Material:madera
Precio:13.000000
=====
Mesa: 2
Alto:14
Ancho:14
Fondo:14
Material:plastico
Precio:14.000000

```

- Ejemplo calculando pedidos:

```

1- Introducir características de una nueva mesa
2- Mostrar las mesas disponibles
3- Calcular precio pedido
4- Salir
3
Hay 3 mesas
Introduzca la lista de índices de mesas separadas por comas sin espacios. Maximo 5 mesas
0,2
El precio final calculado es: 26.000000

```