

- 1) Escribir una función que reciba como parámetros una frase y que retorne cuántas palabras tiene esa frase. Aplicarla en un programa donde el usuario ingrese una frase y luego informe cuántas palabras tiene la frase.
- 2) Escribir una función que ponga en mayúscula la primera letra de cada palabra de una frase (pasada como parámetro). Aplicar dicha función en un programa.
- 3) Escribir una función que retorne la longitud promedio de la palabras de una frase que se recibe como parámetro. Aplicar dicha función en un programa.
- 4) Escribir una función que reciba como parámetros una frase y una palabra y que retorne cuántas veces aparece dicha palabra en la frase. Aplicar dicha función en un programa.
- 5) Escribir una función que reciba como parámetros dos frases y que determine si ambos textos son idénticos. Aplicar dicha función en un programa.
- 6) Programar el juego de cartas "La guerra", bajo las siguientes reglas:
 - a. El mazo contendrá 48 cartas (4 palos numerados del 1 al 12).
 - b. Se mezclará el mazo convenientemente.
 - c. Habrá 2 jugadores que reciban 1 carta cada uno por mano.
 - d. Ganará la mano aquél jugador que tenga la carta de mayor número (no importa el palo) quien sumará 1 punto.
 - e. En caso de empate, los puntos se acumularán para el ganador de la siguiente mano.
 - f. El juego finalizará cuando se hayan repartido los 48 naipes (24 manos). Luego se imprimirá el resultado final del partido.

Utilizar funciones en todos los aspectos posibles y eficientemente.

- 7) Programar el juego de "craps" bajo las siguientes reglas:
 - a. Se lanzan 2 dados.
 - b. Si la suma de los mismos es 7 u 11, el jugador GANA.
 - c. Si la suma es 2, 3 ó 12, el jugador PIERDE.
 - d. Si la suma es otro número, el jugador volverá a tirar los dados indefinidamente hasta que salga 7 (en cuyo caso PIERDE) o hasta que repita la misma suma que en el primer tiro (en cuyo caso GANA)
- 8) Programar un juego con baraja francesa (A, 2, 3, 4, 5, 6, 7, 8, 9 10, J, Q, K – cuatro palos –) con las siguientes reglas:
 - a. Juegan 5 jugadores.
 - b. Cada uno recibe 5 cartas.
 - c. Ganará la mano el jugador que tenga la escalera más alta (no considerar escalera 10, J, Q, K, A pero sí A, 2, 3, 4, 5) que sumará 5 PUNTOS.
 - d. Si ninguno de los jugadores tiene escalera, ganará 1 PUNTO quien tenga la figura más alta (J, Q o K). En caso de empate, todos los jugadores empatados sumarán 1 punto.
 - e. En caso de no haber figuras, ningún jugador sumará puntos.
 - f. Ganará el juego el jugador que llegue primero a los 50 PUNTOS.

Cada mano se mostrarán las cartas de todos los jugadores, qué jugador ganó y cuántos puntos ganó por esa mano y luego los puntajes parciales de todos los jugadores. Al final, se imprimirá qué jugador ganó.

Solución del ej. 1:

```
#include <stdio.h>
#include <conio.h>

int es_separador(char); /* Función booleana para detectar separadores */
int contar_palabras(char []);

int contar_palabras(char frase[])
{
    int i=0, cantidad=0;
    int hay_palabra; /* Variable testigo para detectar si se encontró una
palabra */
    while(frase[i]!='\0')
    {
        /* Se saltean espacios al principio de la frase */
        while(es_separador(frase[i]) && frase[i]!='\0')
        {
            i++;
        }

        hay_palabra=0;

        /* Recorre el arreglo aumentando el subíndice cuando no hay separadores
*/
        while(!es_separador(frase[i]) && frase[i]!='\0')
        {
            i++;
            hay_palabra=1;
        }

        /*Si se detectó una palabra, se suma 1. De lo contrario, se suma 0*/
        cantidad+=hay_palabra;
    }
    return cantidad;
}

int es_separador(char x)
{
    return (x==' ' || x=='.' || x==',' || x=='!' || x=='?' || x==';' || x==':');
}
```

//Falta realizar el main con la carga del texto, queda como ejercicio para el alumno