Ejercicios del Tema 6b

Funciones con otros tipos de datos

1. Copia de ficheros

Escribe un programa que copie el contenido de un fichero en otro cambiando los espacios en blanco por un guión bajo `_`. Además, debe contar cuántos cambios se han producido. Emplea el fichero `lorem_ipsum.txt` como ejemplo. El ejercicio debe resolverse empleando una función con el siguiente prototipo:

```
int espacio_por_guion(char texto[]);
```

2. Cuenta letras

Escribe un programa que analice la frase del ejercicio anterior contando en número de veces que aparece cada vocal, de manera similar al ejercicio del tema 4 de cadenas de caracteres. Al final del programa, se debe pintar en el terminal, para cada vocal, una línea que contiene el mismo número de asteríscos que veces aparece esa vocal. Este programa debe estar programado empleando dos funciones, **cuentaLetra** y **pintaAsteriscos**. Puedes usar el fichero `lorem_ipsum.txt` como ejemplo.

3. Máximo, mínimo y promedio de variables

El fichero `aranjuez.csv` es un fichero que almacena valores numéricos en formato CSV (valores separados por comas). Contiene 4 columnas (variables) y 2898 filas (registros). Cada fila corresponde a un valor diario de una variable meteorológica registrada en la estación localizada en Aranjuez. Estas variables son: temperatura ambiente, humedad, velocidad del viento, radiación solar.

Escribe un programa que lea este fichero almacenando el contenido de cada columna en un vector. A continuación, el programa debe calcular el valor máximo, mínimo y promedio de cada vector, **empleando una función separada para cada cálculo.** Finalmente, el programa mostrará el resultado en pantalla y escribirá estos cálculos en un fichero con un formato similar al siguiente:

Variable	Min	Max	Media
Temp	XX	XX	XX
Humedad	XX	XX	XX
Viento	XX	XX	XX
Rad	XX	XX	XX

4. Nóminas

Se debe calcular la nómina mensual de un empleado que trabaja por horas, el pago de cada hora trabajada depende de su categoría:

categoría	pago x hora (€.)
A	30.00
B	20.50
C	18.50

Además, si el empleado trabaja más de 150 horas mensuales tiene una bonificación del 5 % de nómina. El programa que calcula la nómina está compuesto por dos funciones, una que solicita los datos del empleado por teclado y otra función que calcula la nómina mensual. El programa principal debe mostrar la nómina mensual calculada del empleado introducido.

Para los datos del empleado se debe usar la siguiente estructura:

```
typedef struct
{
    char nom[40], categoria;
    int horas;
    float nomina, pHora, bonf;
} empleado;
```

Los prototipos de las funciones que se tienen que utilizar son los siguientes:

```
empleado pedirDatos(empleado emp);
float calcularNomina (empleado);
```

Ejemplo de ejecución:
Introduzca nombre empleado: Pepe Campo
Introduzca categoría del empleado: B
Introduzca horas trabajadas: 25
La nómina total del empleado Pepe Campo es: 512.50 euros, con un total de 0 euros de bonificación extra.

5. Texto

Prepárese un programa en lenguaje C que solicite una frase al usuario, la modifique y presente en pantalla el resultado, con las siguientes características:

- Se fijará el tamaño máximo de la frase con una directiva define.
- Tendrá una función específica, int modificaT(char texto[]), que realizará la modificación.
- La modificación consistirá en:
  - Solicitar al usuario una letra.
  - Se solucida al usuario una frase
  - Localizarla en el texto y sustituirla por un 5 si está en mayúsculas y por un 7 si está en minúsculas.
- El retorno de la función será el número de sustituciones realizadas.
- La presentación de la frase modificada se hará desde main e incluirá el número de sustituciones realizadas. Si no hay modificaciones solo aparecerá el correspondiente mensaje.
- Finalmente se preguntará al usuario si desea repetir el proceso con una nueva frase y se procederá en consecuencia.

6. Almacén

Se desea hacer un programa informático que calcule la cantidad de dinero inmovilizado en un almacén de productos. El dinero inmovilizado equivale al precio por unidad del producto multiplicado por la cantidad de productos disponibles en el almacén. Además, si la cantidad de productos en el almacén es menor o igual a 10 unidades, debe indicar la necesidad de hacer el pedido de dicho producto.

Estructure su código de la siguiente manera.

Generación del Almacén: vector de estructuras cuya dimensión se establece al inicio del programa según las necesidades del usuario.

Formato del tipo de dato:

```
typedef struct
{
    char cod[5];  //Código del prodcuto
    int stock;    //cantidad de producto en almacen
    float precio; //precio por unidad
}INV;
```

Inicialización del Almacén: se realiza mediante la función generaInventario(). Esta función recibe como parámetros el vector de estructuras y la cantidad de elementos del vector. El usuario debe introducir por teclado cada uno de los elementos de la estructura a fin de inicializarla:

```
void generaInventario(INV inventario[], int numero_items);
```

Gestión del Almacén:: se realiza mediante una función que recibe como argumento un elemento (solo UNO) del vector de estructura previamente inicializado y calcula la cantidad de dinero inmovilizado en el almacén, en euros. Además, avisa al usuario si se necesita hacer un perido cuando el stock sea menor o igual a 10. La función retorna la cantidad de euros inmovilizados del producto:

```
float calculaInmovilizado(INV producto)
```

7. Audio

Un fichero de texto está formado por las muestras obtenidas de la digitalización de una señal de un sensor. La primera fila contiene una cadena de caracteres que representa el nombre de la señal, la segunda fila un número entero que es el número de muestras, y el resto de filas son los números reales que representan las muestras. Elabore un programa que lea el contenido del fichero y almacena los datos en una variable cuyo tipo será el declarado en la siguiente estructura.

```
typedef struct
{
    char nombre[50];      //Código del prodcuto
    int n_muestras;       //cantidad de producto en almacen
    float muestras[100];  //precio por unidad
}audio;
```

Genere un vector de muestras datos.txt con los datos explicados en el enunciado y 10 muetsras de datos para probar su programa.