Práctica 16. Estructuras y ejercicios de repaso

1. Ejercicios propuestos

1.1. Ejercicio

Realiza un programa al que se le pase como argumento el nombre de un archivo con información sobre puntos que forman triángulos. Cada línea del archivo contiene 3 puntos separados por punto y coma (cada punto son 3 coordenadas separadas por espacio en blanco). Si no se ejecuta con el número de argumentos correctos ha de dar un mensaje de error. El programa debe leer el contenido del archivo e indicar qué puntos del triángulo con mayor área y los del triángulo con mayor perímetro.

Estructuras a utilizar y campos que las forman:

- `Punto`: 3 coordenadas reales.
- `Triangulo`: 3 puntos, el valor del área y el valor del perímetro.

Funciones a utilizar:

- `calculaPerimetro`: recibe 3 puntos y devuelve un valor real. Calcula el perímetro de un triángulo formado por esos 3 puntos.
- `calculaArea`: recibe 3 puntos y devuelve un valor real. Calcula el área de un triángulo formado por esos 3 puntos.
- `leerDatos`: se le pasa el nombre de un archivo y el número de líneas. Devuelve un puntero a una estructura triángulo.
- `numLineasArchivos`: se le pasa el puntero a un archivo y devuelve cuántas líneas tiene.
- `mayorArea`: se le pasa un puntero a una estructura triángulo y el número de triángulos que hay, devuelve el índice del triángulo cuyo área es la mayor.
- `mayorPerimetro`: se le pasa un puntero a una estructura triángulo y el número de triángulos que hay, devuelve el índice del triángulo cuyo perímetro es el mayor

Ejemplo de archivo:

```
1 2 3;3 0 0;3 5 0
2 2 0;3 2 0;3 5 0
0 0 0;3 0 0;10 15 0
```

1.2. Ejercicio

Realice un programa en lenguaje C que tenga las siguientes funciones las cuales realizarán las tareas especificadas:

- La función `main` recibe como argumentos dos números y realiza las siguientes tareas:
  - Debe comprobar que el número de argumentos del programa es adecuado. Si no lo es mostrará un mensaje de error. NOTA: El número de argumentos adecuado sería ejecutar el programa como `./programa 3 5`
  - Si el número de argumentos es adecuado, deberá inicializar una matriz con números aleatorios entre 3 y 7. Para ello deberá llamar a la función `inicializa`
  - Después de inicializar la matriz, calculará la media de todos sus elementos. Para ello llamará a la función `media`.
  - Por último imprimirá el resultado por pantalla.
  - A continuación leerá una cadena de cómo máximo 100 letras, incluído el `'\0'` y un entero que corresponderá con una posición. Si el entero no es un número entre 1 y 99 dará un mensaje de error y volverá a leerlo hasta que sea correcto.
  - Llamará a la función `sacaletra` que devuelve la letra de la cadena anteriormente leída que ocupa la posición especificada.
  - Imprimirá la letra devuelta por la función.
  - Fin del programa
- La función `inicializa` recibe como argumentos una matriz y su tamaño e inicializa los elementos de la matriz con valores aleatorios entre 3 y 7. No devuelve nada.
- La función `media` recibe como argumentos una matriz y su tamaño y devuelve la media del valor de los elementos de la matriz.
- La función `sacaletra` recibe como argumentos una cadena y un entero que representa la posición de una letra. Devuelve la letra que ocupa la posición indicada. Es decir, si la cadena es `Esto es un examen` y la posición es `3`, devuelve la letra `t`.

1.3. Ejercicio

¿Qué imprime este programa?

```
#include <stdio.h>
int funcion (int* b, int v[2], int a);
int main (){
  int a, *b, e[2]={1,2};
  a=4; b=&a;
  e[0]=funcion(&a, e, e[1]);
  printf("a=%d, b=%d, e[0]=%d, e[1]=%d", a, *b, e[0], e[1]);
  return 0;
```

```
}
int funcion (int* b, int v[2], int a){
  *b=7;
  v[0]=5;
  a=20;
  return *b;
}
```