Tarea 13:

Fecha de entrega: ***ANTES** del 12 de Noviembre.*

Toda la tarea se debe de hacer usando memoria dinámica (`malloc()` y `free`()) cuando sea requerida memoria.

a) Usando el cargado desde un archivo de texto de un escenario 2D (tarea anterior), programar el laberinto de una gota de agua que "cae con gravedad" en un escenario de plataformas 2D. Este problema que se debe de ejecutar graficamente en la consola se explicó en clase. Algunas notas abajo:

- La gota aparace al inicio en alguna parte del la parte más alta de la pantalla.
- La gota cae si no hay ninguna plataforma abajo de ella.
- Si choca con una plataforma, primero avanza a la derecha buscando un hoyo por donde caer. Si choca con una pared vertical invierte su búsqueda hacia el lado izquierdo. Cada vez que choca con una pared vertical invierte la dirección horizontal de búsqueda (derecha-izquierda).
- La gota sale del laberinto si encuentra el último hoyo en la parte inferior de la pantalla.

b) Hacer un programa que lea la matriz **A** y el vector **b** que se encuentran en archivos de texto en:
https://www.cimat.mx/~alram/elem_comp/mat_A.txt
https://www.cimat.mx/~alram/elem_comp/vec_b.txt
El formato de los archivos es el mismo de la tarea anterior (el vector solo tiene al inicio del archivo el número de elementos). Multiplicar **A***b (información en https://es.wikibooks.org/wiki/Álgebra_Lineal/Matriz_por_vector) y guardar el vector resultado en un archivo **binario** con el mismo formato (longitud en `int` y luego datos en `float`).

c) Hacer un programa que lea el archivo binario que se guardó en el

inciso b) y que muestre su contenido en pantalla.

d) Usando memoria dinámica, hacer un programa que cree una matriz de `doubles` de 100 renglones y 1000 columnas y la llene de números aleatorios en el rango [0,99]. Hacer una función que recibe la matriz como parámetro y que la escribe a un archivo binario (igual que antes, escribe `nr, nc` primero, que son `int` y luego las entradas de la matriz que son `double`) . La función debe de escribir un renglon completo cada vez (es decir, haciendo a lo más 100 llamadas a `fwrite()` para escribir todas las entradas de la matriz).

e) Ejecutar este programa que lee líneas completas en archivos de texto y verificar que funciona y que lo entienden.
https://www.cimat.mx/~alram/elem_comp/main_fgets.c
e.1) Modificar el programa en el inciso e) de tal forma que abra los archivos PLY que se indican al final de esta tarea y que usando `fgets()` y `sscanf()` sea capaz de guardar en variables de tipo `int` y reportar automáticamente en pantalla:
- Cuántos vértices tiene el objeto 3D
- Cuántas caras tiene el objeto 3D
- Cuántas caras son de 3 vértices (es decir, cuantos triángulos componen al objeto).

Es suficiente que su programa funciones con estos 2 archivos PLY:
https://www.cimat.mx/~alram/elem_comp/cube_ball.ply
https://www.cimat.mx/~alram/elem_comp/
Cyl_1p0diam_60_uni_ondul_3cic_1p4ampX_1p4ampY.ply

Lo cuales se ven así, respectivamente:
https://www.cimat.mx/~alram/elem_comp/blender1.jpg
https://www.cimat.mx/~alram/elem_comp/blender2.jpg