Systems Programming: Practical
Pointers and Dynamic Memory Allocation

This practical will provide a few simple exercises to help you understand memory allocation better.

A Dynamic memory allocation

Add to the triangle code you created in Practical 3 so that you can allocate and de-allocate individual triangles using `malloc()` and `free()`.

Modify your program further so that it dynamically creates a pointer-based list of five triangles as they are read from file and then de-allocates the triangles once they have been used to create the output file.

(For this part, you may use a pointer array that is not dynamically allocated, e.g. using something like `struct triangle *T[5];`)

B Recursive triangle subdivision

One way to subdivide a triangle is to draw a line from one vertex to the mid-point of the opposite edge, giving two smaller triangles.
Write a recursive algorithm that will subdivide a triangle a predetermined number of times (i.e. to a given depth); you may choose which side of the triangle to subdivide each time.

Use dynamic memory management to allocate the triangles as they are created and to de-allocate the parent when they are split to form two new triangles.

Hints:

1. To find the point mid-way between two other points you simply need to add the points together and take the average (divide by 2). Given a point (x, y) and a point (p, q) you can form a point midway between them as $(\frac{x+p}{2}, \frac{y+q}{2})$.
2. For this part, you may want to dynamically allocate a pointer array using e.g. `struct triangle **ret = calloc(sizeof(struct triangle*), number);`. (The type `struct triangle **` is a pointer *to a pointer* to a `struct triangle`.) To avoid having to keep track of how many elements you have in your array, you can allocate space for an extra pointer element and set it to `NULL`. Think about how you can combine two arrays of pointers.

C Reading

Ensure you are familiar with the use of pointers described in Chapter 10 (particularly 10.1) `http://www.eskimo.com/~scs/cclass/notes/sx10a.html` and Chapter 11 on memory allocation: `http://www.eskimo.com/~scs/cclass/notes/sx11.html`.