Question 1.

Download `LL.c` from the course website. Read through and understand the functions that are given in the file.

Now, complete the functions where code is missing of the linked list part of `LL.c`. Test the functions. **We will not check off untested code. Testing and debugging is part of the work.** It is always better to debug functions one at a time rather doing it at the end.

Question 2.

In this question, you will implement`ArrayList`, an alternative to linked lists.

The idea is to expand the memory block used to store the data if you are running out of space. You can use `realloc` `https://en.cppreference.com/w/c/memory/realloc` to get a new pointer to an area of memory which has the requested size, and to which the old contents would be copied if necessary. You can (and must) use `memmove` to move the contents of large blocks of memory.

`ArrayList` would keep track of the size of the current list (i.e., the number of elements in it) as well as the capacity (i.e., the maximum possible size).

When appending, you should expand capacity by a factor of 2 using `realloc` if you are running out of space.

Part (a)

Implement and test the `ArrayList` functions.
We will not check off untested code. Testing and debugging is part of the work. It is always better to debug functions one at a time rather doing it at the end.

Part (b)

Read the documentation for `memove` and `memcpy`. Why is `memcpy` an inappropriate choice when implementing `AL_insert` and `AL_delete`?