Building:

To build just type make when you are in the directory.
Type make clean to remove the object and executable files.
My makefile is structured so that it only rebuilds objects when they have been modified.
The final executable is created by linking the objects compiled in the other steps.
The executable will be created with the name “prog”.

Text files:

I created a *read_textfile* function as described in the assignment.
This function can read text files where the words are separated by newlines and spaces.
Many words can be on one line if they are separated by spaces.

When many words are on one line it tracks the first and last cell created for this line, so that the pointer chain can be maintained properly.

It can also handle files that contain blank lines or have many spaces between words.
It does this by using the *strtok* function to tokenise the string by newlines and spaces.

I also created a *write_textfile* function. This is used in the tests to generate a file with some expected contents that can be read by the *read_textfile* function.

You are free to use whatever text file you wish, but make sure to update the expected words in the test function (ensure the order is correct when traversing backwards).

Binary files:

The first part of my binary file contains:

- The number of cells in the list (4 bytes)
- The longest word size in the list (4 bytes).

This is then followed by the following data for each cell:

- The length of the cell’s word (including null terminator);
- The word including the null terminator.

I store the number of cells so I can easily stop reading from the file when I know all cells have been read.

I store the longest word size so I can use a single buffer and easily read words from the file. Without this I would need to ensure the current buffer can hold each word and resize the buffer if not.

To test the binary file functions I simply create a list with some known words, write it to a binary file, and then create a list using that binary file. The order of the words is checked using both the prev and next fields.