To compile *cd* to the directory and type *make*.
The executable will be named *list_tests*

List code:

Some points of note:

My reverse function works by traversing the list and swapping the next and prev fields of each cell. Once this is done the head and tail fields of the list are swapped. This reverses the list without any string copying required.

I created a *find_cell_with_word* function that finds the cell that contains the given word. This is used in both *insert_after* and *insert_before*, so they do not contain duplicate code for this action. This function is static as it should only be used internally by the list code.

I also created a *create_cell_with_word* function that handles creating a new cell and setting the word. This is used by *append*, *insert_after* and *insert_before*, again meaning that they do not need duplicate code to perform this task. This function is also static as it should only be used internally by the list code.

My list functions have some error handling. For example if you use *insert_after*, but the list does not contain the provided word a warning will be printed and the list will be returned unmodified.

Test approach:

I use an array of function pointers to hold all of the test functions.
This array is iterated over and each test function is called.
Each test functions returns 0 if the test passed, and 1 if it failed.

Most of the test functions create a list and fill it with words. These tests then create a string array which contains the words expected to be in the list, in the expected order.

The *compare_expected_and_list* helper function is used to check if the order of the words in the given list matches the order of the words in the given string array.

In addition the *compare_expected_and_list* function will check for unexpected list ends, and unexpected extra entries.

Finally the *compare_expected_and_list* function accepts a direction enum which tells it which way to traverse the list. This allows us to test that all fields (head, tail, next and prev) have been set correctly.