Assignment 6

To compile and run
g++ starter.cpp bits.cpp prime.cpp -o starter && ./starter

Description
I have created test cases for each of the six types of sets in starter.cpp. I have also called all these test cases in the main function. To grade, I recommend looking at the code for each test case and comparing it with the (messy) output.
The test cases are named as follows:

carray_simple_set_test();
carray_range_set_test();
hashed_simple_set_test();
hashed_range_set_test();
bin_search_simple_set_test();
bin_search_range_set_test();

For hashed_simple_set_test(), I have created a hashtable using the (k*F(e) % p) collision policy. p is a special prime greater than the maximum size of the set. The code for computing the special prime is taken from https://www.geeksforgeeks.org/smallest-special-prime-which-is-greater-than-or-equal-to-a-given-number/. The time for insertion and removal is amortized constant. Mostly we insert without collisions, which precludes finding another empty slot using the collision policy.

For bin_search sets, the lookup complexity is logarithmic.
Inserting an element larger than every element that is already in the set has constant complexity, since we know the largest used index in the set, and we simply add the new element to one greater than that index, and then update the largest index.
Otherwise, the insertion complexity is linear. We find the location in logarithmic time but we have to shift all the latter locations to the right by one, which takes linear time.

I have also implemented split functions for the range set, which are called when we remove a range or an element that falls within a preexisting range. They split the relevant range into two and make space for the one extra range by shifting to the right by one all the ranges that are greater.

I have also implemented a comparator for char*'s called "char_pointer_comp" as suggested in the assignment prompt, which can be found at the top of starter.h. It has a bug, however. If you uncomment the commented code in bin_search_range_set_test() that uses this comparator, you will find a segmentation fault.