Data Structures Project: Trie

What has been implemented:

I have implemented the trie data structure, also known as a prefix tree. It has been applied to make a spell-checker that supports all 26 English alphabets plus apostrophes. Each node of the trie has 27 child pointers to other trie nodes, where each node corresponds to a letter or an apostrophe.

It also has the ability to prefix-search i.e. search for all words beginning with a given prefix. For example if the user types "ca", it can return all words beginning with "ca" in the dictionary. It can also support prefix-based deletion.

The main spell-checker works as follows: First a dictionary in the form of a newline separated .txt file is loaded into the trie. I found the dictionary from https://github.com/dwyl/english-words

Then a sample text can be spell-checked against the dictionary by checking if every word in the sample text belongs in the dictionary or not. A wrong spelling is reported when a word is found that is not in the dictionary.

Testing:

I have tested the code within the main functions themselves by having many checks on the kind of input that can be given and I also have assert statements that will prevent the program from running further if an unexpected input somehow makes it way there.

But I also have two of my own functions that test the robustness of the code.

The first inserts a whole dictionary into the trie then deletes it.

The second inserts a whole dictionary into the trie then deletes words beginning with a given prefix.

How to run program:

Compile using:

```
 gcc -o trie trie.c
```

Run using :

```
 ./trie [DICTIONARY] [TEXT]
```

For example

```
./trie dictionaries/custom.txt texts/small.txt
```

We can see the results of the program ourselves by using the custom.txt dictionary and the small.txt since the output of these are easily seen on the terminal, but the same thing happens on a larger scale with the proper words.txt dictionary and the other texts provided.

Since the words.txt dictionary contains numbers and hyphenations, they are picked up as errors by my program due to the given specification of just letters and apostrophes.