Testing on my own system, it takes around 20us to append or remove a sentence, although these numbers go up considerably if the array must be expanded. The time taken to search depends on the position of the target sentence, but seems to stay around 15us – 20us. Deletion is very expensive in my setup, taking 60us just for a small scale test. This is likely due to combination of the search time together with the memory shifting required.

Client algorithm: very simple, split the input string at the first space, if the first word matches with a command, execute that command. Then wait for the server to return a response. If an additional text result was expected, print it out.

Server algorithm: All of the sentences are stored as an array of character pointers. Sentences are appended by adding its pointer to the end of the array. Sentences are removed by searching each string in order until a match is found, and then shifting all other sentences in the array forward. Word deletion is accomplished by searching each sentence for instances of the target word. For each one found, the substring before the word but after the last found instance is concatenated to the new version of the sentence. Then the sentence is replaced by this new version. Word searching is done by tokenizing the sentence into words (by splitting at whitespace and punctuation), and then checking word by word.

The test results are as expected for my data structure, which is just a variable sized array of strings.