The way I performed my parsing and dynamic memory allocation was like this:

I have a character array buf[] that I store the initial input from stdin as a global variable. I also have a global 2D array **commandcomp. My goal is to parse the characters from buf[] into individual strings stored one by one in **commandcomp. The challenging part was correctly dynamically allocating memory for the length of **commandcomp and then additionally the length of each string in **commandcomp.

First, I had a while loop to detect whether there were multiple spaces in a row. If there were, I exit the program as per the directions. Then, I had a second while loop, where I again count spaces to determine the length of **commandcomp and malloc for it. I also add a null pointer to the end of **commandcomp. Then I had another while loop where I count the length of each word and malloc space for it within **commandcomp. For my next and last while loop, I go through character by character and place it in the correct row and column of **commandcomp.