One of the most common string processing problems is the ’parsing’ problem: taking a string passed to you as input and extracting the different input components from that string. In an upcoming Programming Assignment you are going to need to a parse a string containing an operating system command line.

Write a funciton in C/C++ that has the following signature:

char \*\* parseCommandLine(string aCommandLine);

This function takes the input parameter and splits that string up into an array of component tokens that are returned to the caller. As this is assumed to be a Linux command line (for example, from the terminal application), you need to be able to handle certain special cases: (1) file redirection with the ’<’ and ’>’ characters, (2) file redirection with the <<, and (3) the pipe is the only other character used as a seperator. character |. Otherwise, you may assume that the any white-space character is a separator.

Your function needs to return a pointer to a dynamically allocated array of C-strings (remembering that a double-pointer in C++ is another way to say array of something). The seemingly odd return type is required by some of the system calls you will use in your next assignment.

**An Important Note:**

There are multiple ways to solve this problem using the string class in C++.

If you do apply some Google-Fu and research this question, do make certain you understand what you are being asked to do and avoid blindly copying what you find. Not only because of the academic honesty issues but noting that you need to make certain you correctly address the problem requirements.