Program Execution

If you try to type something into your shell that isn't a built-in command, you'll get a message that the shell doesn't know how to execute programs. Modify your shell so that it can execute programs when they are entered into the shell. The first word of the command is the name of the program. The rest of the words are the command-line arguments to the program.

You can assume that the first word of the command will be **the full path to the program**. So instead of running `wc`, you would have to run `/usr/bin/wc`. In the following parts, you will implement support for simple program names like `wc`. However, you can pass some autograder tests by only supporting full paths.

You should use the functions given in `tokenizer.h` for separating the input text into words. You do not need to support any parsing features that are not supported by `tokenizer.h`. When your shell needs to execute a program, it should fork a child process, which calls one of the functions from the `exec` family to run the new program. **You may not use** `execvp` (see Path resolution). The parent process should wait until the child process completes and then continue listening for more commands.

Once you implement this functionality, you should be able to execute programs.

```
./shell
0: /usr/bin/wc shell.c
    77      262     1843 shell.c
1: exit
```
