Minishell Questions

1.)
Q: Why are **exit** and **cd** handled as built-in commands and no child processes are forked? What would happen if you forked children for these commands?

A: **Exit** and **cd** are directly handled by the shell because it is unnecessary to fork and create a child process. It is much easier for the user to exit the shell without creating a child process, and the same goes for cd, as doing so will cause efficiency problems. Specifically, if you forked the exit command, the parent process would be left running, and the whole purpose of using exit is to stop all of the processes, which would be completely useless if it was forked. Likewise, handling cd with forks would change the directory of the child process and not the parent process, and would not affect any of the other processes occurring at the moment.

2.)
Q: Why should you use **killpg**(**getpid**(), **SIGTERM**); before **main**() exits?

A: To start, the function itself **killpg**(**getpid**(), **SIGTERM**) will send the **SIGTERM** signal to the pg, or process group, to "politely" ask it to terminate the process. I say politely due to the fact the **SIGTERM** signal can be "blocked, handled, and ignored" as GNU.org describes it. In reference to main() it is important to use the **killpg** function due to the fact that it will clear out all of the child processes when the user types in **exit**, and once again, when the function is called, it will terminate the program smoothly.