In order to run this file you must first type make to create the shell. Then run ./ish to run the shell. You can now type any bash function as well as these custom functions cd, exit, gcd, args and acker. Acker is ackermann's function, it takes 2 arguments and recursively solves the problem. This program may never finish as ackermann's is recursive and impossible to compute depending on the inputs.

sample : acker 1 4
warning this function may never finish
do you want to continue?[y]: y
6

The process I took to solve this assignment was through lecture and exploration. I first set out to understand the core concepts of an os and tried to plan out the assignment accordingly. The hardest part for me was the background programs because i thought we needed to use sigset(). I ended up reading the man and found it was obsolete and used waitpid() instead. I also ran into the problem of having zombie programs after i exit my shell. The way i overcame this problem was by saving any background PID's to a file named '.pids'. I assume this file will be allowed to be created and will stay in the directory till the end of the program. I also assume that you guys won't be calling './ish &' in an ish shell because that kinda messes with the killing of processes. Also using the parser you guys supplied to us does not properly parse things in quotes i.e args "hello world" this will count as argc = 2 and args hello world which is different from the supplied example. I don't see how this would be possible to do unless we change the lex file.