**Jacob Rook**

**Data Structure and Algorithms II**

**Project 1**

**User’s Manual**

**Setup and Compilation:**

1. Download and unzip the submission from eLearning on a Linux box in the multi-platform lab.
2. The includes:
   1. Main.c
   2. Functions.c
   3. Functions.h
   4. Commands.txt
   5. Makefile
   6. Functional\_Decomposition.docx
   7. UsersManual.docx
   8. README.txt
3. Environment: This program was tested on a combination of the Eclipse and jGrasp IDE. However, also works in the schools ssh server.
4. Compiling. This program includes a Makefile. At the command line in Linux, type make. The program produces an executable: rook-j-p1

**Running the program:** A commands.txt file is included with the input I used to test my project. If the user wants to use their own commands file, simply overwrite the commands.txt file. The name of the file is hard coded into the project to read from commands.txt. No command line arguments are required or checked.

User input: No user interaction with the program is required.

**Output:** All output goes to the console. Output will be similar to this:

$ ls

$ mkdir d1

d1

$ ls

D d1

$ mkdir d3

d3

$ ls

D d1

D d3

$ mkdir d2

d2

$ ls

D d1

D d2

D d3

$ pwd

rook/root/

$ cd d2

rook/root/d2/

$ ls

$ mkdir a1

a1

$ ls

D a1

$ pwd

rook/root/d2/

$ cd a1

rook/root/d2/a1/

$ ls

$ cd d1

d1 is not located in a1

$ cd ..

rook/root/d2/

$ pwd

rook/root/d2/

$ cd ..

rook/root/

$ ls

D d1

D d2

D d3

$ cd d1

rook/root/d1/

$ mkdir a1

a1

$ ls

D a1

$ pwd

rook/root/d1/

$ cd ..

rook/root/

$ cd..

Tokens[0] "cd.." was not recognized

$ addf File3

File3

$ addf File2

File2

$ addf File1

File1

$ ls

D d1

D d2

D d3

F File1

F File2

F File3

$ mv File2 File4

File4

$ ls

D d1

D d2

D d3

F File1

F File3

F File4

$ rm File1

$ ls

D d1

D d2

D d3

F File3

F File4

$ rm d1

$ ls

D d2

D d3

F File3

F File4

$ whereis a1

rook/root/d2/a1/

$ bye