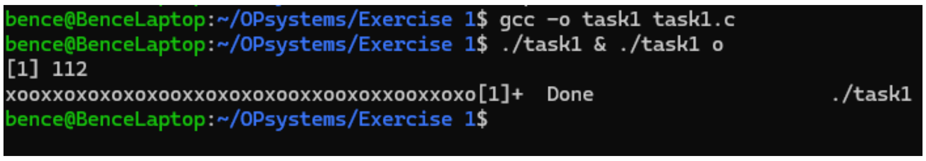
Operating Systems Project - Exercise 1

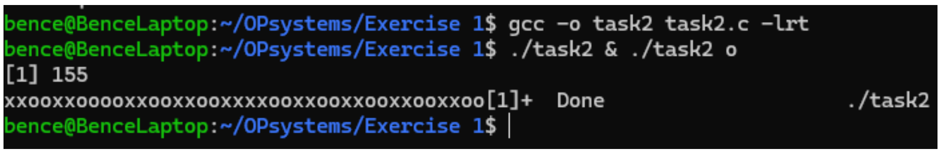
Task 1



```
bence@BenceLaptop:~/OPsystems/Exercise 1$ gcc -o task1 task1.c
bence@BenceLaptop:~/OPsystems/Exercise 1$ ./task1 & ./task1 o
[1] 112
xooxxoxoxoxoxooxxoxoxoxooxxooxoxxooxxoxo[1]+  Done                    ./task1
bence@BenceLaptop:~/OPsystems/Exercise 1$
```

This output tells us a few things about mutual exclusion; since both instances of the file access stdout concurrently, we see mixed x-s and o-s. The rand()%3 and rand()%2 introduce a random factor to the sleeping time, which is why we cannot see a pattern in it.

Task 2

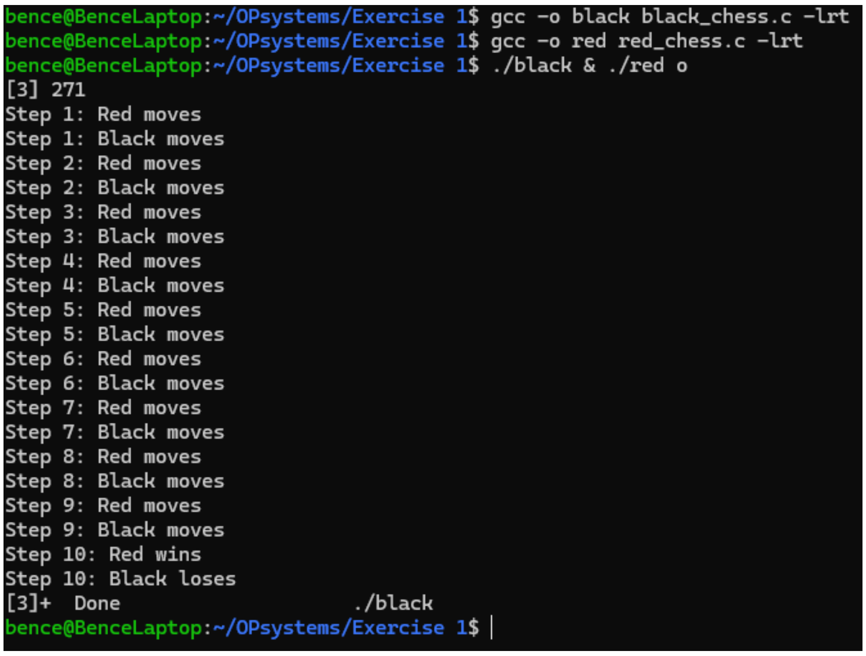


```
bence@BenceLaptop:~/OPsystems/Exercise 1$ gcc -o task2 task2.c -lrt
bence@BenceLaptop:~/OPsystems/Exercise 1$ ./task2 & ./task2 o
[1] 155
xxooxxooooxxooxxooxxxxooxxooxxooxxooxxoo[1]+  Done                    ./task2
bence@BenceLaptop:~/OPsystems/Exercise 1$ |
```

This output tells us how semaphores are used to control access to the output steam. In the code, it regulates so that only one process can print at a time by using sem_wait to decrease the semaphore's value and lock the section before printing and sem_post to increase it and unlock the section. This is why the outputs are printed in pairs.

Task 3

This code simulates a game which is being played back and forth, by red posting "hei" semaphore and waiting for "hong" semaphore, and black posting "hong" semaphore and waiting for "hei" semaphore. This simulates the turn based gameplay of chess, which we can see is successful from the output. Each 'player' does 10 steps, with the last one being the end of the game where we can see that the red side won and the black side lost. The randomized processing time in the code simulates how in a real chess game, the players would take variable length time thinking on their moves. After the last move, each program closes and unlinks the semaphores.



```
bence@BenceLaptop:~/OPsystems/Exercise 1$ gcc -o black black_chess.c -lrt
bence@BenceLaptop:~/OPsystems/Exercise 1$ gcc -o red red_chess.c -lrt
bence@BenceLaptop:~/OPsystems/Exercise 1$ ./black & ./red o
[3] 271
Step 1: Red moves
Step 1: Black moves
Step 2: Red moves
Step 2: Black moves
Step 3: Red moves
Step 3: Black moves
Step 4: Red moves
Step 4: Black moves
Step 5: Red moves
Step 5: Black moves
Step 6: Red moves
Step 6: Black moves
Step 7: Red moves
Step 7: Black moves
Step 8: Red moves
Step 8: Black moves
Step 9: Red moves
Step 9: Black moves
Step 10: Red wins
Step 10: Black loses
[3]+  Done                    ./black
bence@BenceLaptop:~/OPsystems/Exercise 1$ |
```