Part 1: Exec Family System Call and Basic IPC using Signals API

- ➔ The 3 files are **q1.c**, **E1.c**, & **E2.c** as required by the problem statement.
- ➔ **q1.c** first creates process **S1** then **ST** and **SR** in main().
- ➔ **S1** registers **SIGTERM** with handler **sigterm()** defined in **q1.c** and **setitimer** which runs the process for 2 seconds.
- ➔ Processes **ST** and **SR** call functions **SR()** and **ST()** with the pid of S1 as arguments.
- ➔ **ST()** and **SR()** have **execv()** system calls with paths for executables **E1** and **E2** and pid of **S1** as arguments.
- ➔ **E1** registers signal **SIGALRM** with handler sigalrm defined in **E1.c**.
- ➔ **E1** signals **SIGALRM** which enqeues a random number to the shared memory of all 3 files.
- ➔ **E1** signals **SIGTERM** to **S1** which prints the random number.
- ➔ **E2** registers signal **SIGALRM** with handler sigalrm defined in **E1.c.**
- ➔ **E2** signals **SIGALRM** which finds CPU timestamp count and interprets it then sends it to the shared memory of all 3 files.
- ➔ **E2** signals **SIGTERM** to **S1** which prints the timestamp.

Instruction to Run:

1. cd to **Q1** folder
2. run **make** in terminal.
3. run **./q1**