Question 2

Components:

1) Main Code (1.c):
    a) This is main code for the program. It manages the creation of 3 child processes, and running a separate C program with different scheduling policy.
    b) It measures the execution time of the processes and writes the result in file “output.txt”.
    c) Child processes run the Counting Program named “2.c”.
2) Counting Code (2.c):
    a) It performs a long for-loop counting for 2^32.
3) Plotting Code:
    a) It plots a graph, using the data of file “output.txt”, which shows the time taken by each scheduling process.

Code Flow:

1. The code is executed using a BASH file named “simple.sh”.
2. Which first compiles the Counting Code and generates the executable file.
3. And then the compilation of Main Code begins and generates the executable file.
4. I have made 3 nested if-else statements for executing to run the child processes made by using fork().
5. The Child Processes calls for another program i.e. “2.c” using different scheduling policies (by using `sched_setscheduler())` as per given in question, and their priority order was fixed (Others to 0; FIFO to 1; RR to 1).
6. By using the `clock_gettime()` function before calling another program and after the execution of the program, we calculate the time taken for execution, and we can write the time taken by each process in the file “output.txt”.
7. And at last after waiting for all the child process, to complete their execution, the main program ends.
8. Then the graph is plotted using the data of the “output.txt”, which shows the time taken by each scheduling processes.
9. We are running all the processes on single core processor (by using `taskset --cpu-list ./my_program` command), so the final order of execution time of processes is **FIFO<RR<OTHERS**.