**Documentation**

All the executable programs can take an argument of a directory of a text file if there is no argument it will runs the defaults text file. (e.g. ./a file)

I added a new Enum NOTREADY to initialize the process when saving into the struct. The struct of the process has also added a deadline since is we also need to read it from the input file.

**Assumption**

The process is input in the order by the entry time.

**Task 1**

A program that uses first come first serve scheduling to go through the program. In this task the process that first entry will run, and the second process will check if the previous process is finished or not. If not, it will continue runs the previous process. This will be ended only if the last process is exited.

**Task 2**

A program that uses round robin scheduling. In this task it will also runs first process which entered and will save the time quantum in the struct to check if the process runs for the specific time. Beside I added WAITS and QUEUE Enum for waits is to put the finished time quantum process to a wait state to see any new process need to entry and QUEUE is to not repeat add process to the queue.

**Task 3**

A program that will give most deadline met algorithm in this task I used shortest remaining time to schedule the processes. Besides if there is same remaining time process I implement to check if the deadline of the process and the current time to see which one has the lesser time then it will run that one.