REPORT ON THE ASSIGNMENT 1
by CS16BTECH11039

The program implements the discrete event simulation for rate monotonic scheduling and for earliest deadline scheduling .
The events considered for the program are the following:
1.when process starts it's execution
2.When process is preempted
3.when process finishes
4.when resumes its execution

A process strucutre is created that contain the processid,period,deadline which is equal to period as per the question,the process time , the number of times the process will execute, the waitTime, the start time of the process which is 0 for all processses at the start of the program.

A class CPU is created to represent an actual like CPU which can hold the process . The Cpu calss has it's own clock which is represented by the 'ticks' variable in the program.It has a file stream to log the events that occurs during the simulation. It also holds the ready-Cum-Event queue for both the algorithms.

** The ready-cum-event queue is used to procure events as well as the process that will be scheduled on the cpu **

The CPU has rmProcessing and edfProcessing methods to act like real processing of the process it has a while loop which increases the clock. It checks for any possible preemption using isPreempted method if it founds any preemtion then it returns the control to the edf/rm scheduler. If the process completes it's process time then it is released from the Cpu using the updateProcessInCpu() method
and logs the event.

Two seperate funcitons edfSchedule and rmSchedule are used to depict the edf and rm schedulers.Both the methods use the same design.

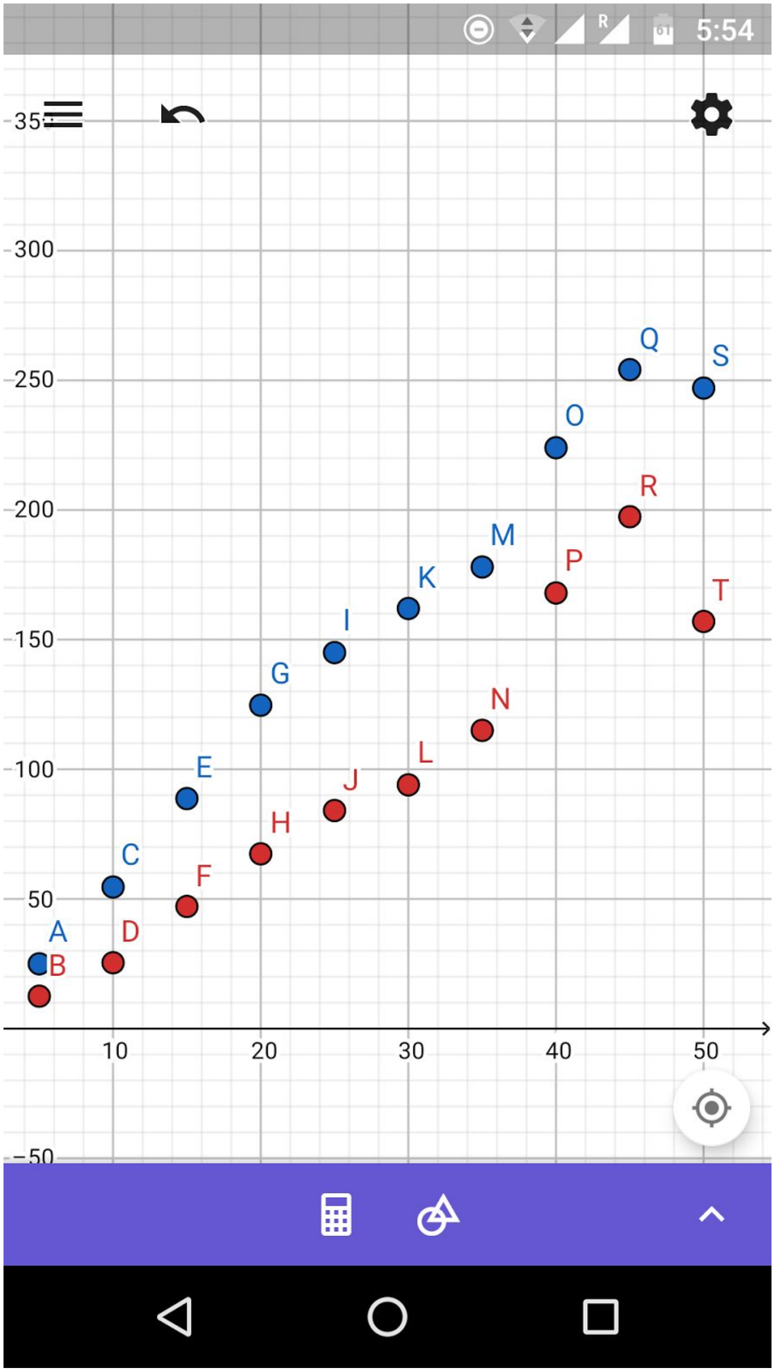
The rmSchedule() method first sends the process in the front of the readyqueue to the cpu using rmProcessing() method. The after it’s completion it runs a while loop to check if there is any event that occurs in the ready-cum-event queue if yes then it gets the event which will take place using the getEvent() function . After that the schedulable process is sent for processing using rmProcesssing() method. If there is a schedulable process but no event which will take place at the current time then the cpu is ordered to stay idle using cpuIdle() method.

***The context switch time is included in the program and the time take is 1s as per the question so when a process completes it’s execution or preemted at t sec then another process starts it’s processing ar t+1 sec. ***

The preempt(),complete(),resume(),processIt() methods are just used to log the events in the Log file associated with the CPU.

The problem which I encountered when creating the program was the design of the event and ready queue which i decided to combine and the passing of control from one method to another.While writing the scheduler algorithm the upadtion of the deadlines and the preemtion part were a bit difficult.Also to schedule the older process which didn’t get any process time lead to some problems at the begining.

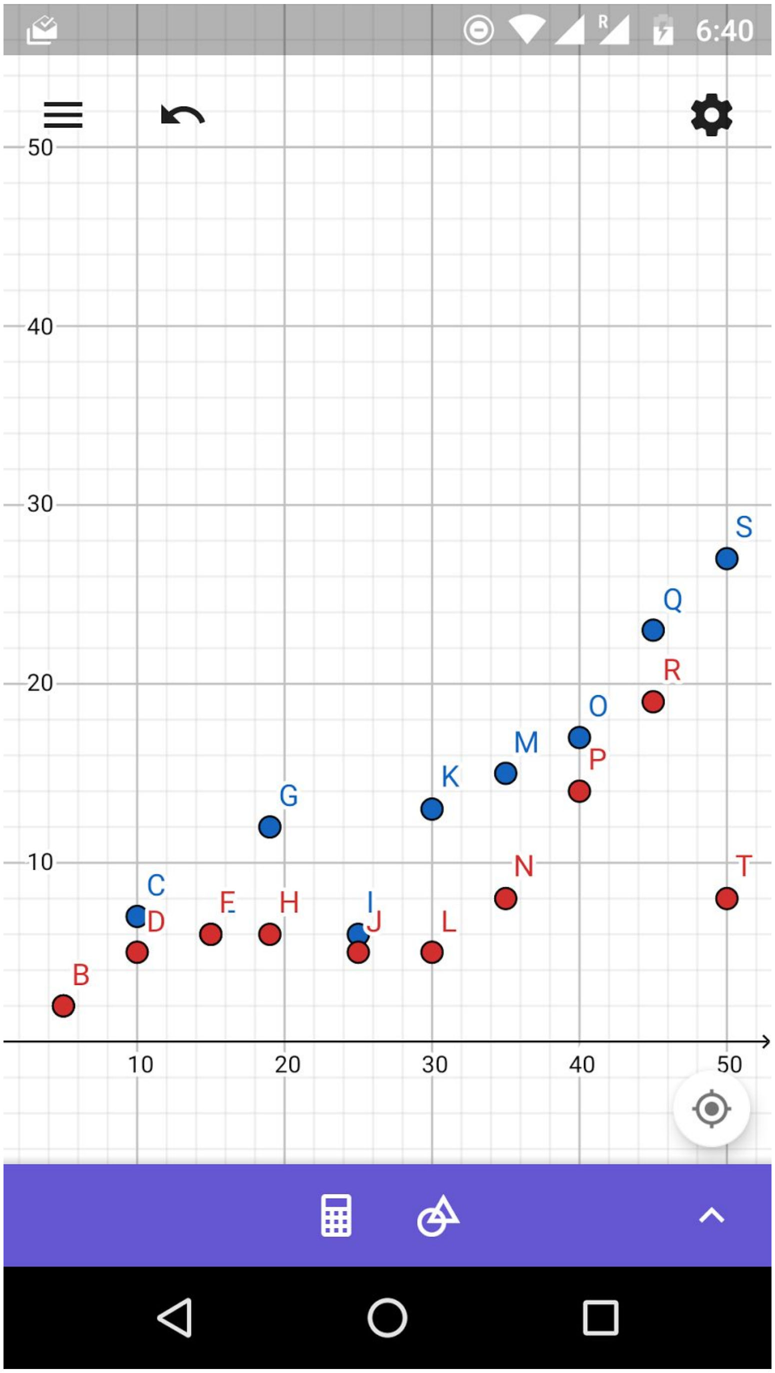
The graph for the AVG wait time vs number of processes is given below.



The red dot is for the EDF and blue dot for RM scheduling.

As one can infer from the graph that EDF scheduling provides less AVG wait time than RM scheduling.

The graph for the number of deadlines missed vs number of processes is the following



From the above graph one can infer that the EDF algorithm provides much better reduction in missed deadlines than RM scheduling.
The red dot is for EDF and blue dot is for RM scheduling.

EDF allows full process utilization and exploits better computational resource than rate monotonic scheduling. Provides better responsiveness than rate monotonic scheduling