NO IPC in this assignment

Given a set of processes, it was the task of the operating system to assign the programs to different memory regions, so that they could be executed concurrently. Assume that you have a system with the following memory specifications:

- The total memory size is 1024B.
- Each process size is less than or equal 256B
- Assume single uni-core CPU

You are going to write the “Buddy System Allocation” algorithm to allocate memory to different processes requests which are scheduled using Round-Robin scheduling algorithm.

Input

- Read the processes from the “input.txt” file which has the following format:

```
Quantum         20
Switch          2
process_id      run_time        arrival_time    mem_size
T1              30              0               200
T2              60              10              100
T3              30              15              110
T4              30              20              170
T5              2               72              50
```

- The first line contains the round robin algorithm time quantum. It’s composed of 2 words “tab separated”: “Quantum” and any positive integer variable value; in our case it’s 50, but it can be any other value, 5 for example.
- The second line contains the **process switching time.** It’s composed of 2 words ”tab separated”: “Switch” and any positive integer variable value; in our case it’s 1, but it can be any other value, 10 for example.
- The third line is the processes table header. It’s composed of 4 words ”tab separated”: “process_id”, “run_time”, “arrival_time” and “size” which is an integer representing the process required memory to run.
- Each line starting at the fourth line till the end, is composed of 4 words “tabs separated”: a string process id and 3 positive integer variable values representing the run time, arrival time and memory size for this process respectively.

Requirement

- Schedule the process using the round robin algorithm.
- When a process is scheduled for running, allocate the size required by this process using the “Buddy System Allocation” algorithm:

- Assume an overhead of 6B on each process (i.e. if a process size is 50B, then it will occupy 56B on the memory).
- If there is enough space it should run directly
- If there isn't enough memory, place it at the end of the queue
- If it has been placed at the end of the queue more than 5 times, halt the job (i.e. remove it from the queue and tell the user that there wasn't enough space for it)
- Since RR may schedule a long process multiple times due to the fact that a process is stopped once the quantum is finished; assume that the process memory need remains unchanged in each time it's rescheduled by the RR algorithm, and it remains in memory until it's finished even if it's in the queue and not running.

Output

2 files, an output.txt file and a log.txt file

- Output.txt should look like this:
  - Where the mem_start indicates the starting address of the process address space and mem_end indicates its end
  - Feel free to add the scheduling statistics at the end of the file (e.g. avg. turnaround time, avg waiting time, avg. response time). It's optional.

process_id	run_time	arrival_time	finish_time	mem_size	mem_start	mem_end
T1	30	0	109	200	0	255
T2	60	10	191	100	256	383
T3	30	15	150	110	384	511
T4	30	20	167	170	512	767
T5	2	72	153	50	384	447

- log.txt should look like this

```
Queue: T1
Executing process T1  : started at 0, stopped at 20, 10 remaining memory starts at 0 and ends at 255
Process switching     : started at 21, finished at 23
Queue: T2, T3, T4, T1
Executing process T2  : started at 24, stopped  at 44, 40 remaining, memory starts at 256 and ends at 383
Process switching     : started at 45,finished at 47
Queue: T3, T4, T1, T2
Executing process T3  : started at 48,stopped  at 68,10 remaining , memory starts at 384 and ends at 511
Process switching     : started at 69,finished at 71
Queue: T4, T1, T2, T3
Executing process T4  : started at 72,stopped at 92, 10 remaining, memory starts at 512 and ends at 767
Process switching     : started at 93, finished at 95
Queue: T1, T2, T3, T5, T4
Process switching     : started at 96, finished at 98
Executing process T1  : started at 99, finished at 109, memory starts at 0 and ends at 255
```

```
Queue: T2, T3, T5, T4
Process switching     : started at 110,finished at 112
Executing process T2  : started at 113,stopped at 133, 20 remaining, memory starts at 256 and ends at 383
Process switching     : started at 134,finished at 136
Queue: T3, T5, T4, T2
Process switching     : started at 137,finished at 139
Executing process T3  : started at 140,finished at 150, memory starts at 384 and ends at 511
Queue: T5, T4, T2
Executing process T5  : started at 151,finished at 153, memory starts at 384 and ends at 447
Queue: T4, T2
Process switching     : started at 154,finished at 156
Executing process T4  : started at 157,finished at 167, memory starts at 512 and ends at 767
Queue: T2
Process switching     : started at 168,finished at 170
Executing process T2  : started at 171,finished at 191, memory starts at 256 and ends at 383
```

Development Environment:

- OS choice: you can use either Windows or Linux.
- Language choice: you can use whatever you like

Regulations:

- Due to: Monday, Dec 18th, 11:59 pm (open to discussion with ur representative in case of delivery conflicts)
- Delivery Rules:
  - Add a text file to your project folder named "group_members.txt" containing your team names and the student codes.
  - Add a Readme.txt file explaining how to run your code and what tools/OS you used.
  - Compress the whole project and give it a name, example: bahgat_adel_dina.zip
  - Remove the executable files from your folder as some mail servers filters executable files, so the mail will never reach us.
  - Send an email to: d.tantawy@gmail.com [or elearning if exsist]
    - Subject: [OS][Credit][Ass] Team no.
    - Don't forget to attach the compressed file
- Cheating is prohibited, and is graded ZERO for all parties.
- Cheating includes copying from internet, from your friends or older students writing for you.
- Late deliveries are penalized by losing 10% of the phase grade for each late day.
- This is a group assignment.
- Students who had low participation in the first assignment have to work hard here or they will get a zero grade
- Violation of any of the above regulations will be penalized