National University of Computer and Emerging Sciences, Lahore Campus

Course:	Data Structures Lab	Course Code:	CL218
Program:	BS(Computer Science)	Semester:	Fall 2022
Duration:	1:30 hours	Total Marks:	100
Paper Date:	24th Oct, 2022	Weight	25%
Section:	BCS-3D	Page(s):	2
Exam:	Midterm	Roll No:	

Instruction/Notes:

- **We will check your code for plagiarism.** If plagiarism is found, it will result in F grade in lab.
- **In case of any ambiguity make suitable assumption.**
- No cell phones are allowed. Sharing of **USBs** or any other items is **not allowed.**
- You are not allowed to have any helping code with you.
- Submission path is **\\cactus1\Xeon\Fall 2022\Fareeha Ashfaq\DS 3D\Mid\ D1/D2** (Submit your code in your respective section).
- Make different **.cpp files** for both questions named as roll#_Q# (**21L-1234_Q1**)
- Make **folder** with their roll# (**21L-1234**) and places all files (**both .cpp + .txt file**)
- **Don't** submit the .zip folder.

Question 1: **(60 marks)**

A CPU can execute many processes. However, at a given time, a CPU can execute the instructions of only one process (not true for modern CPUs, nonetheless). To manage the execution of n processes, we make use of what is known as "CPU Scheduler". We will write a basic type of scheduler which will work as follows: Suppose we have 10 processes named p1, p2,....,p10. Each process has some number of instructions n1, n2, n3.....n10, respectively. Now to execute all the processes, we first execute some instructions, let's say 3 instructions of process p1, and then we will execute 3 instructions of p2, so on and so forth. After that we will restart from process p1 and execute 3 instructions of p1, then p2, then p3, so on and so forth. Then the cycle begins again. So, the processes are being executed by CPU in circular fashion. If a process has finished its instructions during this cycle, then we remove that process from this cycle. The given example below depicts the scenario:

Suppose we have 3 processes:
Process_id: p1
Total_Instructions: 7

Process_id: p2
Total_Instructions 6

Process_id: p3
Total_Instructions 5

7 6 5

3	3	3	3	3	2	1
1	2	3	1	2	3	1

Scheduler Output:
3 instructions of p1 executed.
3 instructions of p2 executed.
3 instructions of p3 executed.

```
3 instructions of p1 executed.
3 instructions of p2 executed.
p2 has finished execution.
2 instructions of P3 executed.
p3 has finished execution.
1 instruction of p1 executed.
p1 has finished execution.
```

Implement your scheduler function using the queue. The function will take a file name as argument. The file contains information about all the processes to be executed. A sample file is also given. The first number in the file tells about the number of instructions that the CPU will execute, of a process p, at one time. The second number tells us about the total number of processes in the file. Test your function in main with the given file.

```
void scheduler(string processFile);
```

Question 2: **(40 marks)**

You are given the head of a singly linked-list. The list can be represented as:

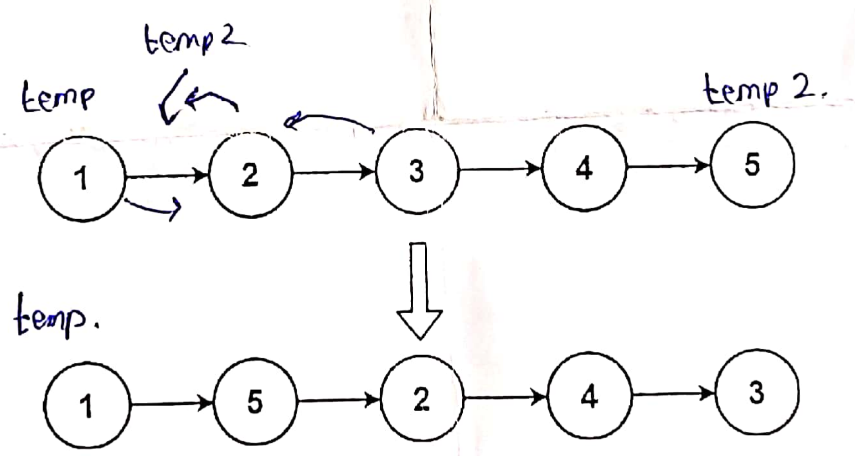
$L_0 \rightarrow L_1 \rightarrow \dots \rightarrow L_{n-1} \rightarrow L_n$

Reorder the list to be on the following form:

$L_0 \rightarrow L_n \rightarrow L_1 \rightarrow L_{n-1} \rightarrow L_2 \rightarrow L_{n-2} \rightarrow \dots$

You may not modify the values in the list's nodes. Only nodes themselves may be changed.

Example:



Input: head = [1, 2, 3, 4, 5]

5/2

Output: [1, 5, 2, 4, 3]

1 2 3 4 5 6 7

1 7 2 3 4 5 6

1 2 3 4

1 4 2 3

1 7 2 6 3 4 5 1 5 2 3 4.