January 2021 CSE204: Data Structures and Algorithms I Sessional

Offline on Linked List

You are probably familiar with the Pillow Passing game. In this game, some people sit in a circle and pass a pillow to the player sitting next to him/her (say to the left or to the right i.e. in a particular direction). Some music plays in the background and when it finishes, the one holding the pillow is eliminated. And then another music begins and the process continues until there is only one player left. The last person left is declared as a winner.

To make the game more interesting, suppose we allow a new player to enter the game at any instance of time. But whoever enters, sits beside the one currently holding the pillow, in the opposite direction of the pillow passing i.e. if the player holding the pillow is supposed to pass it to his right, then the new player will sit beside him on the left.

Besides, since the reflex time of different persons may vary, we assume each person requires a certain amount of time to pass the pillow after receiving it. These reflex times are given as input.

A brief description of the possible events and their identifiers are listed in the following table.

Command	Event Description
M	Current music ends and a new one starts. The player holding the pillow is eliminated and it is passed to the next player. Print the player who has just been eliminated.
R	Reverse the pillow passing direction i.e. the person holding the pillow will start passing it in the opposite direction. (Assume it incurs no additional time)
I x	A new player enters the game whose reflex time is x seconds. But if the game is already won, i.e. only one person is left, the new player can no longer join the game.
P	Print who is currently holding the pillow.
F	End the game and print who is currently holding the pillow, the pillow passing direction and the players still in the game sequentially.

Table 1 : List of Events

Input/Output:

You will take input from the console and print output to the console in an interactive way. That is, you should process each line of input and print the corresponding line of output (if any) as required.

Input Format:

The first line of input will contain a positive integers *N* indicating the number of players who are initially starting the game. The next line contains *N* positive integers denoting their reflex times. Each of the next lines will contain time in seconds (in a non-decreasing order) which indicates the occurrence of an event as specified in Table 1.

The initial players are numbered from 1 to *N* and any other player entering the game later will have subsequent numbers as identifiers. You can assume the game starts from Player 1 and the initial direction of pillow passing is from Player *i* to Player (*i*+1).

Output Format:

Print the outputs of the events as specified in Table 1. See the sample I/O very carefully for further clarification.

Sample I/O:

Input	Output
3 4 5 2 6 P 6 M 6 P 7 P 8 P 9 P 10 I 1 12 P 13 P 15 P 17 P 22 R 23 P 24 P 25 F	Player 2 is holding the pillow at t=6 Player 2 has been eliminated at t=6 Player 3 is holding the pillow at t=6 Player 3 is holding the pillow at t=7 Player 3 is holding the pillow at t=8 Player 1 is holding the pillow at t=9 Player 1 is holding the pillow at t=12 Player 3 is holding the pillow at t=13 Player 4 is holding the pillow at t=15 Player 1 is holding the pillow at t=17 Player 3 is holding the pillow at t=23 Player 3 is holding the pillow at t=24 Game over : Player 1 is holding the pillow at t=25, pillow passing sequence = Player 1, 4, 3
3 4 5 2 6 P 6 M 6 P 7 F	Player 2 is holding the pillow at t=6 Player 2 has been eliminated at t=6 Player 3 is holding the pillow at t=6 Game over : Player 3 is holding the pillow at t=7, pillow passing sequence = Player 3, 1
3 4 5 2 5 R 6 F	Game over : Player 2 is holding the pillow at t=6, pillow passing sequence = Player 2, 1, 3
3 4 5 2 1 M 2 M 3 I 5 5 M 10 F	Player 1 has been eliminated at t=1 Player 2 has been eliminated at t=2 Game over : Player 3 wins!!

Constraint: You must use the Linked List data structure to solve this problem. You have to implement Linked List by yourself, directly using library data structures is not allowed.

Hints:

1. You should process the inputs in an infinite loop (until you get an F command)
2. When you get a command, you should finish the events until reaching its associated time (e.g. in case of 6 P at line 3 of the first sample input, you should process a passing from Player 1 to Player 2 at t=4).
3. You can keep track of the time when the next passing will occur. This may be updated when you get an M command.
4. The direction of passing should be updated when you get an R command.
5. If only one player is left s/he is considered as the winner and no further M, R, I x commands need to be processed (check the fourth sample input).

Special Instructions:

Write ***readable, re-usable, well-structured, quality*** code. This includes but is not limited to writing appropriate functions for implementation of the required algorithms, meaningful naming of the variables, suitable comments where required, proper indentation etc.

Please **DO NOT COPY** solutions from anywhere (your friends, seniors, internet etc.). Any form of plagiarism (irrespective of source or destination), will result in getting -100% marks in the offline. Also, be informed that for repeated offence of plagiarism, the departmental policies suggest stricter measures.

Submission Guideline:

1. Create a directory with your 7 digit student id as its name
2. Put the source files only into the directory created in step 1
3. Zip the directory (compress in .zip format; .rar, .7z or any other format is not acceptable)
4. Upload the .zip file on Moodle.

For example, if your student id is 1805xxx, create a directory named 1805xxx. Put only your source files(.c, .cpp, .java, .h, etc.) into 1805xxx. Compress 1805xxx into 1805xxx.zip and upload the 1805xxx.zip on Moodle.

Failure to follow the above-mentioned submission guideline may result in upto 10% penalty.

Submission Deadline: March 19, 2021 11:55 PM