CS2123 Program #3 Linked List (20 points)

In this program, you will create a linked list for managing timed events. You will simulate arrivals and departures at different "clock" times. You will advance a "clock" based on the next event time.

You have been furnished with an include file and a data file.

Examine the **Person**, **Event**, **NodeLL**, and **Simulation** **typedef**s in the cs2123p3.h include file. You may modify it to use a different representation of a linked list.

Input Data: (use stdin)

There are many records (terminated by EOF) that define arrivals:

szName iDeltaDepart iDeltaNextArrival

11s 3d 3d

Consider this input:

Fred 10 5

Daphne 4 3

Velma 6 6

Shaggy 9 6

Scooby 2 5

Based on that data, we have arrivals at these times:

0 Fred (departs +10)

5 Daphne (departs +4)

8 Velma (departs +6)

14 Shaggy (departs +9)

20 Scooby (departs +2)

25 End of File … no arrival

For this program, there are only two event types (there will be different event types in Program #4):

EVT\_ARRIVE - an arrival event

EVT\_DEPART - a departure event

You are required to create a **runSimulation**(Simulation sim, int iTimeLimit) function which is passed a sim (which points to a clock and a timed event list). It is also passed a time limit which will help prevent infinite loops. Pass in a time limit of 1000 clock units. **runSimulation** will loop through the timed events (which are placed in order by event time in the linked list) and set the clock to the time of the current timed event. Based on the event, (for now) it prints the time, person, and event. In program #4, the events will do more than just print a message.

|  |  |  |
| --- | --- | --- |
| **Time** | **Person** | **Event** |
| 0 | Fred | Arrive |
| 5 | Daphne | Arrive |
| 8 | Velma | Arrive |
| 9 | Daphne | Depart |
| 10 | Fred | Depart |
| 14 | Shaggy | Arrive |
| 14 | Velma | Depart |
| 20 | Scooby | Arrive |
| 22 | Scooby | Depart |
| 23 | Shaggy | Depart |
| 23 | SIMULATION | TERMINATES |

Since there aren't any additional events, the simulation terminates. Also terminate if the clock's time is greater than or equal to the 2nd parameter to **runSimulation**.

Notice that some of the events are at the same time, but **InsertOrderedLL** in the notes **didn't allow insertion of duplicates**. **You must change it.**

Your output:

* Print the time of each event, the person, and the event type (as text).

**Turn in all of your C code, include file, and the output. As usual, your code must be written according to my standards.**