Ferry Loading

Before bridges were common, ferries were used to transport cars across rivers. River ferries, unlike their larger cousins, run on a guide line and are powered by the river's current. Cars drive onto the ferry from one end, the ferry crosses the river, and the cars exit from the other end of the ferry.

There is an *l*-**meter**-long ferry that crosses the river. A car may arrive at either river bank to be transported by the ferry to the opposite bank. The ferry travels continuously back and forth between the banks so long as it is carrying a car or there is at least one car waiting at either bank. Whenever the ferry arrives at one of the banks, it unloads its cargo and loads up cars that are waiting to cross as long as they fit on its deck. The cars are loaded in the order of their arrival and the ferry's deck accommodates only one lane of cars. The ferry is initially on the left bank where it had mechanical problems and it took quite some time to fix it. In the meantime, lines of cars formed on both banks that wait to cross the river.

The first line of input contains *c*, the number of test cases. Each test case begins with the number *l*, a space and then the number *m*. *m* lines follow describing the cars that arrive in this order to be transported. Each line gives the length of a car (in **centimeters**), and the bank at which the car awaits the ferry ("left" or "right").

For each test case, output one line giving the number of times the ferry has to cross the river in order to serve all waiting cars.

Sample input

```
4
20 4
380 left
720 left
1340 right
1040 left
15 4
380 left
720 left
1340 right
1040 left
15 4
380 left
720 left
1340 left
1040 left
15 4
380 right
720 right
1340 right
1040 right
```

Output for sample input

```
3
3
5
6
```

Original problem by : Piotr Rudnicki

This problem can be easily solved using your queue data structure (two of them). You must write your implementation file using a linked list instead of an array as we did in class. You can even make the list "doubly linked" which means that in addition to have a next pointer, every node also has a previous pointer so that you can traverse the list in either direction. This is not required for this assignment but you may find that it is good practice and helps to make some operations easier. Submit your program to the UVA online judge and send me the acceptance email along with your code files. The solution should be submitted to the online judge as problem 11034.

All programs must include a comment section at the top of the program as outlined below:

```
/*************************************************************************
    Program:     <name of program>
    Author:      <your name>
    Date:        <date you finish the program>
    Time spent:  <total amount of time spent on the project>
    Purpose:     The purpose of this program is to blah blah blah
*************************************************************************/
```