In class we have built an abstract data type using an opaque object called MY_LIST. I would simply like for you to use the MY_LIST type to write a program what will solve the following problem by using your list as a queue data structure.

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 arrives ("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

How do you solve it? The trick to this problem is realizing that your list type can behave as a queue by inserting items using tail insertion and processing items from the head. You will need two queues for this problem, one for the left bank and one for the right. If you are feeling industrious you can actually submit your program to an online checker that will test it by going to uva.onlinejudge.org, making an account, and submitting your solution to problem 11034.