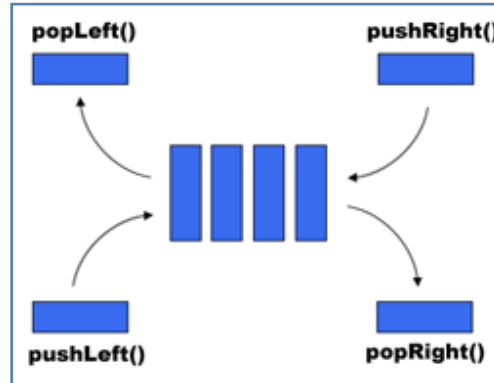


1212 – Double Ended Queue

A queue is a data structure based on the principle of 'First In First Out' (FIFO). There are two ends; one end can be used only to insert an item and the other end to remove an item. A Double Ended Queue is a queue where you can insert an item in both sides as well as you can delete an item from either side. There are mainly four operations available to a double ended queue. They are:

1. **pushLeft()**: inserts an item to the left end of the queue with the exception that the queue is not full.
2. **pushRight()**: inserts an item to the right end of the queue with the exception that the queue is not full.
3. **popLeft()**: removes an item from the left end of the queue with the exception that the queue is not empty.
4. **popRight()**: removes an item from the right end of the queue with the exception that the queue is not empty.



Now you are given a queue and a list of commands, you have to report the behavior of the queue.

Input

Input starts with an integer **T** (≤ 20), denoting the number of test cases.

Each case starts with a line containing two integers **n, m** ($1 \leq n \leq 10$, $1 \leq m \leq 100$), where **n** denotes the size of the queue and **m** denotes the number of commands. Each of the next **m** lines contains a command which is one of:

| | |
|--------------------|--------------------------------------------------------------------------|
| pushLeft x | pushes x ($-100 \leq x \leq 100$) in the left end of the queue |
| pushRight x | pushes x ($-100 \leq x \leq 100$) in the right end of the queue |
| popLeft | pops an item from the left end of the queue |
| popRight | pops an item from the right end of the queue |

Output

For each case, print the case number in a line. Then for each operation, show its corresponding output as shown in the sample. Be careful about spelling.

| Sample Input | Output for Sample Input |
|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 3 8 pushLeft 1 pushLeft 2 pushRight -1 pushRight 1 popLeft popRight popLeft popRight | Case 1: Pushed in left: 1 Pushed in left: 2 Pushed in right: -1 The queue is full Popped from left: 2 Popped from right: -1 Popped from left: 1 The queue is empty |