

### C. LR-remainders

time limit per test: 2 seconds  
memory limit per test: 256 megabytes

You are given an array  $a$  of length  $n$ , a positive integer  $m$ , and a string of commands of length  $n$ . Each command is either the character 'L' or the character 'R'.

Process all  $n$  commands in the order they are written in the string  $s$ . Processing a command is done as follows:

- First, output the remainder of the product of all elements of the array  $a$  when divided by  $m$ .
- Then, if the command is 'L', remove the leftmost element from the array  $a$ , if the command is 'R', remove the rightmost element from the array  $a$ .

Note that after each move, the length of the array  $a$  decreases by 1, and after processing all commands, it will be empty.

Write a program that will process all commands in the order they are written in the string  $s$  (from left to right).

#### Input

The first line contains an integer  $t$  ( $1 \leq t \leq 10^4$ ) — the number of test cases in the input. Then descriptions of  $t$  test cases follow.

Each test case of the input is given by three lines.

The first line contains two integers  $n$  and  $m$  ( $1 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 10^4$ ) — the initial length of the array  $a$  and the value to take the remainder by.

The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^4$ ) — the elements of the array  $a$ .

The third line contains a string  $s$  consisting of  $n$  characters 'L' and 'R'.

It is guaranteed that the sum of the values of  $n$  for all test cases in a test does not exceed  $2 \cdot 10^5$ .

#### Output

For each test case, output  $n$  integers  $b_1, b_2, \dots, b_n$ , where  $b_i$  is the remainder when dividing the product of all elements of the current state of the array  $a$  by  $m$  at the beginning of the execution of the  $i$ -th command.

#### Example

input	Copy
4 4 6 3 1 4 2 LRRRL 5 1 1 1 1 1 1 LLLLL 6 8 1 2 3 4 5 6 RLLRLR 1 10000 10000 R	
output	Copy
0 2 4 1 0 0 0 0 0 0 0 0 4 4 4 0	

#### Note

In the first test case of the example:

- $3 \cdot 1 \cdot 4 \cdot 2 \bmod 6 = 24 \bmod 6 = 0$ ;
- $s_1 = \text{L}$ , so we remove the first element and get the array  $[1, 4, 2]$ ;
- $1 \cdot 4 \cdot 2 \bmod 6 = 8 \bmod 6 = 2$ ;
- $s_2 = \text{R}$ , so we remove the last element and get the array  $[1, 4]$ ;
- $1 \cdot 4 \bmod 6 = 4 \bmod 6 = 4$ ;
- $s_3 = \text{R}$ , so we remove the last element and get the array  $[1]$ ;
- $1 \bmod 6 = 1$ ;
- $s_4 = \text{L}$ , so we remove the first element and get an empty array.

Codeforces Round 927 (Div. 3)

Finished

Practice

Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

Submit?

Language: GNU G++20 13.2 (64 bit, win)

Choose file: Choose File No file chosen

Submit

Last submissions

Submission	Time	Verdict
<a href="#">247135105</a>	Feb/18/2024 22:11	Accepted
<a href="#">247131548</a>	Feb/18/2024 21:34	Accepted
<a href="#">247061031</a>	Feb/18/2024 16:23	Wrong answer on test 2

Problem tags

brute force data structures implementation math two pointers \*1400

No tag edit access

Contest materials

Announcement

Tutorial