

Tasks

task	noga	povez
input	standard input	
output	standard output	
time limit	5 seconds	1 second
memory limit	64 MB	
points	100	100
	200	

NOGA

N grasshoppers are waiting in line to see a show. Waiting is boring, so they waste doing one of the following two moves:

- The grasshopper in position A jumps over B grasshoppers to his left, or
- The grasshopper in position A jumps over B grasshoppers to his right.

The grasshoppers aren't all of the same height. When jumping, a grasshopper must be careful not to hit another grasshopper with its leg. More precisely, it needs to jump as high as the **height of the tallest grasshopper it is jumping over**.

Given the sequence of jumps, output the height of each jump.

Input

The first line contains two integers N and J ($2 \leq N \leq 100\,000$, $1 \leq J \leq 100\,000$), the number of grasshoppers in line and the number of jumps.

The second line contains N integers less than $100\,000$, the heights of grasshoppers in the initial ordering. The first grasshopper is initially in position 1, the second in position 2 etc.

Each of the following S lines describes one jump. Every line contains an integer A ($1 \leq A \leq N$), the position of the jumping grasshopper, a the direction in which it jumps ('L' for left, 'D' for right), and an integer B ($1 \leq B \leq N$), the number of grasshoppers it jumps over. Each jump will be valid (the number B will be less than or equal to the number of grasshoppers on the corresponding side of the grasshopper in position A).

The jumps are given in the order in which the grasshoppers do them.

Output

Output J lines, containing the heights of the jumps, in order in which they are performed.

Example test case

input

```
9 3
5 3 8 4 9 3 7 4 2
2 D 3
8 L 2
5 D 2
```

output

```
9
7
4
```

Clarification: in the first jump, a grasshopper jumps over grasshoppers of height 8, 4 and 9. In the next jump, grasshoppers of height 3 and 7 are jumped over. In the last jump, the heights are 3 and 4.

POVEZ

Mirko and Slavko are playing a game. Mirko is the game master, and Slavko is the player. Slavko has a **headband** over his eyes, so he doesn't see what Mirko is doing.

The game is played with N cups numbered 1 through N . At the start of the game Mirko puts a ball in one of the cups (Slavko doesn't know which one). Then, in one step, Slavko says a letter, 'A' or 'B'. Each time Slavko says a letter, Mirko takes the ball and puts it in one cup (possibly the same cup he took it from). The cup into which Mirko puts the ball is uniquely determined by the current position of the cup and the Slavko's letter. That is, if the ball is in the cup K , it is moved to either cup A_k or B_k (these numbers are predetermined and Slavko knows them).

Slavko's task is to **move the ball into cup 1, regardless of where Mirko put the ball initially**. Write a program which finds any sequence of letters that Slavko has to say in order to move the ball into cup 1. The sequence must be shorter than 10 000 letters.

Input

The first line of input contains the integer N ($1 \leq N \leq 500$), the number of cups.

Each of the following N lines contains two integers, the numbers A_k and B_k for one cup. The first of these lines contains A_1 and B_1 , the second A_2 and B_2 etc.

Note: for input data will be such that the solution, although possibly not unique, will exist.

Output

Output the sequence of A's and B's Slavko should say.

Example test cases

input	input
4	10
1 2	4 9
1 3	5 3
3 1	1 8
1 4	6 1
output	3 8
ABA	9 1
	10 1
	6 3
	2 3
	8 1
	output
	BABAABAB