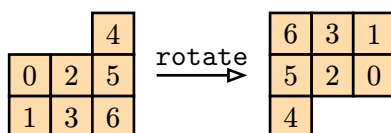


Folding Cloth (folding)

When visiting the Duchfabrik cloth factory museum near the reservoir lake, you are given a strip of cloth modelled as N unit cells labelled $0, 1, \dots, N - 1$ from left to right. Initially, it occupies N distinct columns, with each column containing exactly one cell. Later, you will fold it over several steps until all the cells are stacked in a single column.

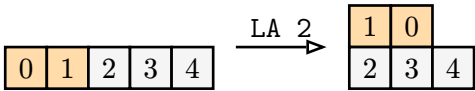
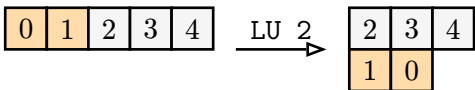
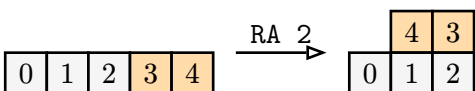
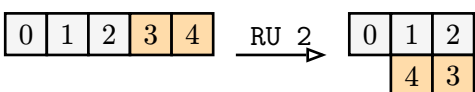
In each step, you fold the cloth as follows. First, take the leftmost or rightmost x columns. Then rotate them 180 degrees as a whole, and place them above or under the remaining cloth. Finally, if this leaves a gap inside a column, press that column's cells together. The precise meaning of these actions is explained below.

To see what “rotate 180 degrees as a whole” means, suppose a block of three columns currently holds the cells on the left below. Rotating the whole block 180 degrees gives the picture on the right:

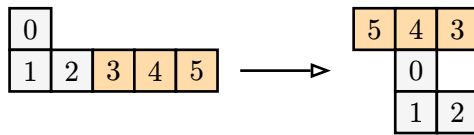


This is a real 180-degree rotation: the columns end up in the opposite left-to-right order, and inside every column the cells end up in the opposite top-to-bottom order.

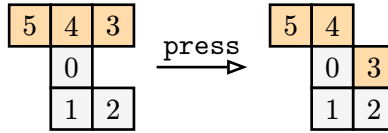
We have four fold types, depending on whether you take the leftmost or rightmost columns, and whether you place them above or under the remaining cloth. In the examples below, the highlighted cells are the ones you fold; the pale cells do not move.

Type	Meaning	Example ($x = 2$)
LA x	Take the leftmost x columns, rotate, and place them above the remaining cloth, aligning the leftmost column.	
LU x	Take the leftmost x columns, rotate, and place them under the remaining cloth, aligning the leftmost column.	
RA x	Take the rightmost x columns, rotate, and place them above the remaining cloth, aligning the rightmost column.	
RU x	Take the rightmost x columns, rotate, and place them under the remaining cloth, aligning the rightmost column.	

Here is a more complex example. Starting with $N = 6$, after doing LA 1 the cloth has $L = 5$ columns, shown on the left below. We now do RA 3: take the rightmost 3 columns, rotate them 180 degrees as a whole, and place them above the remaining cloth. The result before pressing is on the right:



As there's a gap in the third column, we close it by pressing the cells together. It can be shown that pressing up or down doesn't matter in this task. Here we press down, keeping the lowest cell of the column fixed.



This completes the explanation of the precise meaning of these actions.

Now you are given a sequence of M folding steps, each of the form LA x , LU x , RA x , or RU x , applied in order. These steps are guaranteed to result in all N cells being stacked in a single column. Your task is to determine the final vertical ordering of the cells in that column, listed from top to bottom.

Implementation

You will have to submit a single `.cpp` source file.



Among this task's attachments you will find a template `folding.cpp` with a sample implementation.

You will have to implement the following function:

C++

```
vector<int> fold(int N, vector<int> T, vector<int> X)
```

- Integer N represents the number of cells of the cloth.
- M denotes the number of folding steps; it equals the size of both arrays T and X (which always have the same size).
- The array T , indexed from 0 to $M - 1$, contains the fold types: T_i represents the fold type of the $(i + 1)$ -th folding step, where $T_i = 0$ means LA, $T_i = 1$ means LU, $T_i = 2$ means RA, and $T_i = 3$ means RU.
- The array X , indexed from 0 to $M - 1$, contains the fold lengths: X_i represents the value of x for the $(i + 1)$ -th folding step.
- The function must return an array of N integers, representing the final vertical ordering of the N cells from top to bottom.

Sample Grader

Among this task's attachments you will find a simplified version of the grader used during evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the function `fold` and writes back on `stdout` using the following format.

The input is made up of $M + 1$ lines, containing:

- Line 1: the integers N and M .
- Line $2 + i$ ($0 \leq i < M$): the integers T_i and X_i .

The output is made up of a single line, containing the N values returned by the function `fold`, separated by spaces.

Constraints

Let L_i be the number of columns occupied by the cloth after the first i steps. In particular, $L_0 = N$.

- $1 \leq M < N \leq 500\,000$.
- $0 \leq T_i \leq 3$ for each $0 \leq i \leq M - 1$.
- $1 \leq X_i < L_i$ for each $0 \leq i \leq M - 1$.
- The given sequence of steps results in all cells occupying the same column.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- **Subtask 0 [0 points]**: Sample test cases.
- **Subtask 1 [14 points]**: $N \leq 2000$.
- **Subtask 2 [7 points]**: $X_i = 1$ and $T_i = 0$ for each $0 \leq i \leq M - 1$.
- **Subtask 3 [8 points]**: $X_i = 1$ and $T_i \in \{0, 1\}$ for each $0 \leq i \leq M - 1$.
- **Subtask 4 [9 points]**: $X_i = 1$ and $T_i \in \{0, 2\}$ for each $0 \leq i \leq M - 1$.
- **Subtask 5 [11 points]**: $X_i = 1$ for each $0 \leq i \leq M - 1$.
- **Subtask 6 [12 points]**: $2X_i \leq L_i$ and $T_i = 0$ for each $0 \leq i \leq M - 1$.
- **Subtask 7 [21 points]**: $2X_i \leq L_i$ for each $0 \leq i \leq M - 1$.
- **Subtask 8 [18 points]**: No additional constraints.

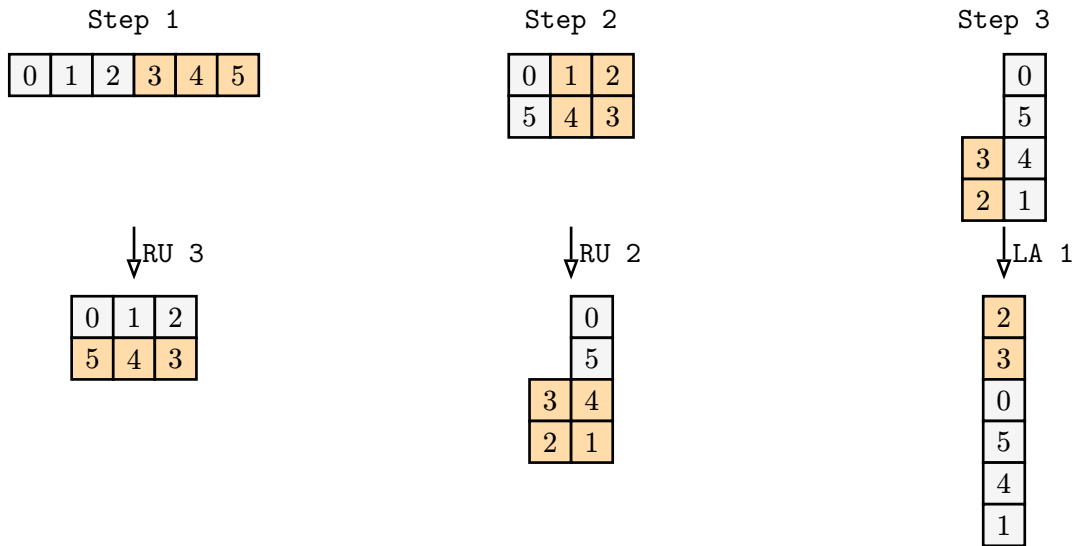
Examples

stdin	stdout
6 3 3 3 3 2 0 1	2 3 0 5 4 1
6 4 0 1 2 3 1 2 2 1	5 3 2 1 0 4

Explanation

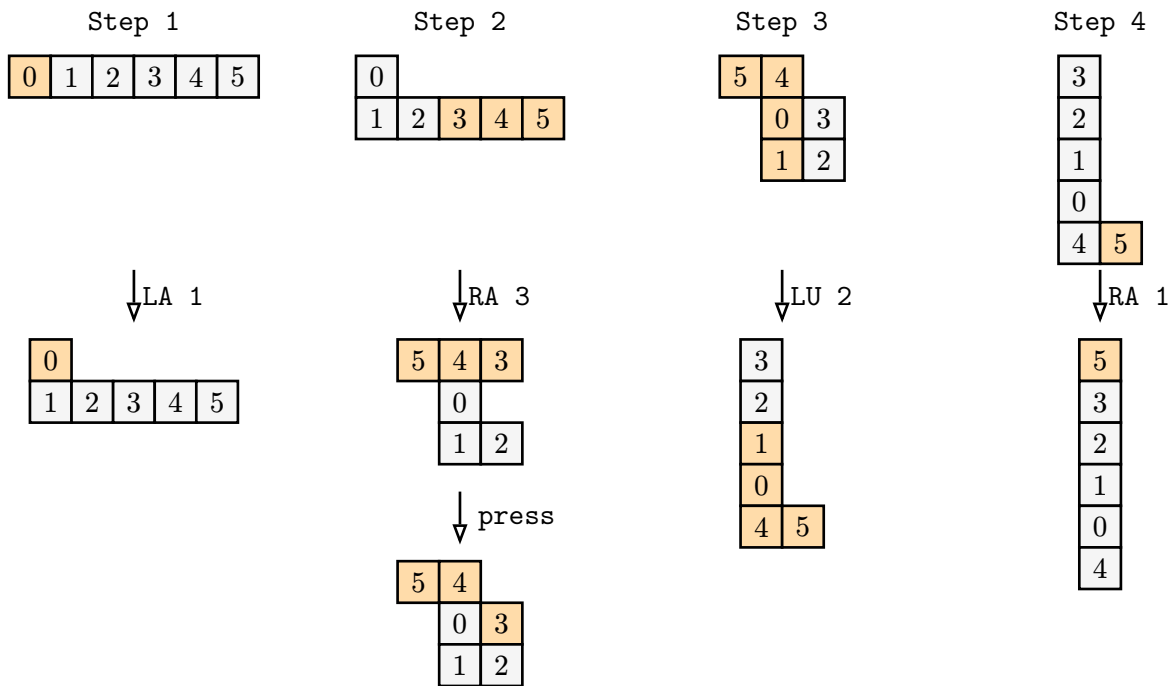
⇒ In addition to the explanations below, you can use the attachment [folding_animation.html](#) to animate any input in the same format as the sample grader.

In the first example, $N = 6$, $T = [3, 3, 0]$, and $X = [3, 2, 1]$, which represents the sequence of steps RU 3, RU 2, LA 1. The cloth evolves as shown below. In each step, the upper drawing repeats the state before the step, the lower drawing shows the state after the step, and the highlighted cells are the cells you fold in that step.



After the third step, all six cells are stacked in one column with the order $[2, 3, 0, 5, 4, 1]$ from top to bottom.

In the second example, $N = 6$, $T = [0, 2, 1, 2]$, and $X = [1, 3, 2, 1]$, which represents the sequence of steps LA 1, RA 3, LU 2, RA 1. The second step requires pressing: we first show the state before pressing, and then show the pressing itself.



After the fourth step, all six cells are stacked in one column with the order $[5, 3, 2, 1, 0, 4]$ from top to bottom.