UNSW ICPC Workshop T3W3
Hard Problem Set
Source: South Pacific Divisionals and ANZAC rounds (various years)

Discuss the problems in this document and try to solve them with your group. Make sure everyone is comfortable with the solution before moving on.
Ask us if you need help, or want to check your solution.

We recommend doing the problems in the given order (roughly difficulty order), but if you don’t like a problem feel free to skip it.

The second, third and fourth problems have links if you wish to code and submit to them

Problem E
Assigning Workstations

Picture by NASA via WikiMedia Commons

Penelope is part of the admin team of the newly built supercomputer. Her job is to assign workstations to the researchers who come here to run their computations at the supercomputer.

Penelope is very lazy and hates unlocking machines for the arriving researchers. She can unlock the machines remotely from her desk, but does not feel that this menial task matches her qualifications. Should she decide to ignore the security guidelines she could simply ask the researchers not to lock their workstations when they leave, and then assign new researchers to workstations that are not used any more but that are still unlocked. That way, she only needs to unlock each workstation for the first researcher using it, which would be a huge improvement for Penelope.

Unfortunately, unused workstations lock themselves automatically if they are unused for more than m minutes. After a workstation has locked itself, Penelope has to unlock it again for the next researcher using it. Given the exact schedule of arriving and leaving researchers, can you tell Penelope how many unlockings she may save by asking the researchers not to lock their workstations when they leave and assigning arriving researchers to workstations in an optimal way? You may assume that there are always enough workstations available.

Input

The input consists of:

- one line with two integers n ($1 \leq n \leq 300\,000$), the number of researchers, and m ($1 \leq m \leq 10^8$), the number of minutes of inactivity after which a workstation locks itself;
- n lines each with two integers a and s ($1 \leq a, s \leq 10^8$), representing a researcher that arrives after a minutes and stays for exactly s minutes.

Output

Output the maximum number of unlockings Penelope may save herself.

Sample Input 1	Sample Output 1

3 5
1 5
6 3
14 6
	2

Sample Input 2	Sample Output 2

5 10
2 6
1 2
17 7
3 9
15 6
	3

Problem F
Interview Queue

Time Limit: 4 second

A very popular position has just opened up at Poodle Inc. Candidates have formed a queue while they wait for their turn to be interviewed.

Competition is fierce and each candidate knows that they will not be selected if another candidate is strictly better than them. Every minute, each candidate looks at the resumé of candidates who are currently adjacent to them in the queue (ahead and behind). If at least one of the neighbour's resumé's perceived value is strictly greater than their resumé's perceived value, they leave the queue since they do not want to waste their time. The candidates all look at their neighbor's resumé simultaneously, and then some candidates leave the queue simultaneously.

This process continues until no more candidates leave the queue. Determine the number of minutes that pass while this process takes place. Report the values of the candidates that leave the queue in each round. Also report the final state of the queue.

Input

The first line of input contains a single integer N ($1 \leq N \leq 100\,000$), which is the number of candidates.

The second line contains N integers $v_1, \ldots, v_N$ ($0 \leq v_i \leq 10^9$ for each $1 \leq i \leq N$), where v_i is the perceived value of the i^{th} candidate.

Output

Display M, the number of minutes taken by this process. Then display M lines. The i^{th} line should contain the perceived values of the candidates who left the queue in the i^{th} minute in the same relative order that they appeared in the queue. Finally display a line indicating the final list of perceived values in the queue after candidates no longer leave it.

Sample Input 1

```
10
3 6 2 3 2 2 2 1 5 6
```

Sample Output 1

```
2
3 2 2 1 5
3 2 2
6 6
```

Sample Input 2

```
3
17 17 17
```

Sample Output 2

```
0
17 17 17
```

Sample Input 3

```
7
8 1 2 3 5 6 7
```

Sample Output 3

```
2
1 2 3 5 6
7
8
```

Network System Administration

Input File: *standard input*
Output File: *standard output*

You are the Network System Administrator (NSA) for a local school. Recently hired on a huge salary after a catastrophic school website defacement, you are responsible for monitoring the school network to keep the systems safe.

The school administration had agreed to give you access to a stream of student e-mail *meta-data* (the time, sender and recipient of every e-mail sent through the network) although this decision has recently come under fire from the student body over privacy concerns.

In order to prove to the school administration that your complete access to the email meta-data is justified, you will write a program to show how the meta-data would be useful in the case of a *trojan mule* attack on the network. From your expert NSA training, you know that a trojan mule is a computer virus a lot like a trojan horse except that it does not reproduce.

In particular, a trojan mule starts on some initial computer then attaches itself to the first e-mail sent from that computer. Upon transmission to the e-mail's recipient, the virus removes itself from the sender's computer and infects the recipient, then waits for the next outgoing e-mail to attach to and so on. Thus, only one copy of the virus exists in the network at any time.

Your program must be able to answer queries of the form ”if a trojan mule was initally on computer X, what computer would it be on now?”. These queries may be interspersed with new e-mail meta-data.

Input

- The first line of input will contain one line with two space-separated integers N and M, representing the number of computers and the number of input lines.
- The following M lines of input will each take one of the two forms:
  1. `E` x y ($1 \leq x, y \leq N$ and $x \neq y$), indicating a new e-mail sent from computer x to computer y; or,
  2. `Q` x ($1 \leq x \leq N$), a query of the virus' current host given that it was initially on computer x.

Output

Your program must output one line for each query (`Q` x line of input), containing one integer representing the computer that the virus would be on at that point in time if it started on computer x.

(continued over . . .)

Sample Input

```
5 8
E 1 2
E 3 4
Q 4
E 2 3
E 3 5
Q 1
E 5 2
Q 2
```

Sample Output

```
4
5
2
```

Explanation

There are 5 computers connected through the school network.

1. An email is sent from 1 to 2.
2. An email is sent from 3 to 4.
3. We are asked, if the trojan mule was initially on computer 4, where is it now? As no emails have been sent from 4 yet, the trojan mule would still be on 4.
4. An email is sent from 2 to 3.
5. An email is sent from 3 to 5.
6. We are asked, if the trojan mule was initially on computer 1, where is it now? The trojan mule was transmitted from 1 to 2 (#1) then from 2 to 3 (#4) then from 3 to 5 (#5) so it would now be on computer 5.
7. An email is sent from 5 to 2.
8. We are asked, if the trojan mule was initially on computer 2, where is it now? The trojan mule was transmitted from 2 to 3 (#4) then from 3 to 5 (#5) then from 5 back to 2 (#7) so it would now be back on computer 2.

Subtasks & Constraints

For all subtasks, $1 \leq N \leq 100,000$ and $1 \leq M \leq 100,000$.

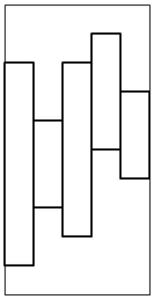
- For Subtask 1 (10 points), $1 \leq N \leq 100$ and $1 \leq M \leq 100$.
- For Subtask 2 (25 points), $1 \leq N \leq 100$ and $1 \leq M \leq 50,000$.
- For Subtask 3 (30 points), no computer receives or sends any more e-mails after it has sent one e-mail. That is, once a computer appears as an x in an `E` line, it will not appear in any future `E` lines.
- For Subtask 4 (35 points), no further constraints apply.

House of a Thousand Blades

Input File: *standard input*
Output File: *standard output*

Seiko is a student of the *katana* (sword). She practises from morning to night under the watchful eye of her teacher, an old man who offers her deep wisdom hidden in cryptic sayings.

Today her teacher takes her to the *koi* pond and produces a flat wooden board. He carefully paints a design on the board: a series of vertical stripes of equal width. The right edge of each stripe touches the left edge of the next, with the whole design forming one contiguous shape that spans the width of the board. One such masterpiece is shown to the right.



The teacher then explains to Seiko today's lesson:

> "Elegance. The swordsmaster wastes neither time nor effort. Each movement of your blade must be definite and imbued with purpose. Take your sword and carve out the shape I have painted on this board, so that it becomes a single detached piece."

Seiko's swordwork is precise, and in one cut she can carve through any straight line segment of the board, regardless of the angle of the line segment or whether its endpoints have already been cut.

Meditating on her teacher's words, Seiko realises that her task now is to find the smallest number of cuts she needs to carve out the painted shape.

Input

The first line consists of two space separated integers, W and H ($1 \leq W, H \leq 100\,000$), representing the width and height of the wooden board respectively. The board can be thought of as a rectangle with co-ordinates ranging from (0,0) to (W, H).

The second line consists of W integers describing the top edges of the W brush-strokes. Each height will be between 1 and H inclusive.

The third line consists of W integers describing the bottom edges of the W brush-strokes. Each height will be between 0 and $H - 1$ inclusive. It is guaranteed the top edge of every brush-stroke is strictly higher than its bottom edge, and that all the brush-strokes form one connected shape, i.e. for all $i > 1$, $bottom_i < top_{i-1}$ and $bottom_{i-1} < top_i$.

For 40% of the available marks, $N \leq 1\,000$.

Output

You should output one line with one integer – the minimum number of cuts needed to obtain the desired shape from the board.

Sample Input

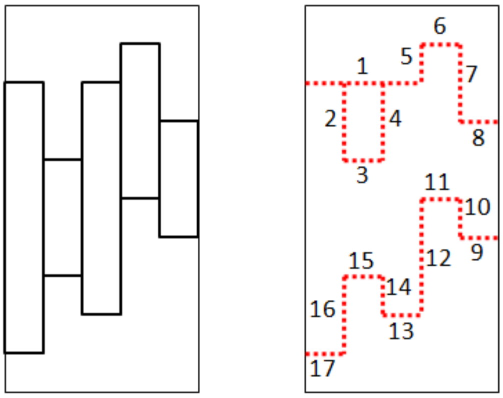
```
5 10
8 6 8 9 7
1 3 2 5 4
```

Sample Output

```
17
```

Explanation

The sample data corresponds to the board shown below on the left. The 17 cuts are shown on the right.



Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.