Spring 2022 mBIT Standard Division

May 22, 2022

This contest is themed around our imaginary amusement park, mBIT Land, and you are the main character! A key feature of mBIT Land is it's main street - Blair Boulevard. Good luck and have fun at the park!

- The mBIT Team

P.S. Our art team retired :(

These problems are roughly ordered by difficulty. However, you should read and think about as many problems as you can in the time given. Good luck and happy coding!

Contents

Program Specifications

The memory limit for every problem is 256 MB. The time limit for every problem is 2s.

Advice

Understand the pretests. When you submit code during the contest, it will be run on 10 pretests and your verdict for each pretest will be displayed. The first pretest will always be the sample but you will not be able to access the data for any other pretest. The pretests have varying sizes but are together designed to cover the full range of input.

Understand the system tests. If your code passes all 10 pretests, shortly after the round ends your submission will be judged on 40 system tests. You will be considered to have solved the problem if and only if your code passes all 40 system tests.

Understand the scoring system. All problems are worth the same amount. Your team's score is the number of problems you fully solve. Your team's rank is determined firstly by your score and secondly by the time of your last correct submission. There is no penalty for wrong submissions.

Use fast I/O. For problems with large input sizes, you may want to use fast I/O methods to prevent a time limit error. Here is how to use fast I/O in each language:

- In Python, write `from sys import stdin, stdout` at the top of your program. When reading input, use `stdin.readline()`. To write output, use `stdout.write()`.
- In Java, use a custom Scanner class as shown here.
- In C++, write `ios_base::sync_with_stdio(false); cin.tie(NULL);` at the top of your `main` method. Then you can use `cin` and `cout` as usual. Printing a single newline character (`\n`) is faster than `endl`.

Special considerations for Python. Make sure you're using fast I/O methods as described above. You can increase the recursion limit if your functions use repeated recursion. Additionally, we strongly recommend submitting your solutions in **PyPy**, which is typically faster than Python. Make sure you strip your input of trailing white space (this is especially important for our grader). Finally, `exit()` and `quit()` do not work with our grader.

Language versions. Our grader uses C++ 17 (compiled with gcc 10 using `-Ofast` and `-march=native`), Java 17, and a PyPy version that implements Python 3.7.

Ask for clarifications! If you are confused about a problem statement, do not hesitate to message us. Message us in our discord here.

§1 Ticket Prices

Before entering mBIT Land, your friends Alice, Bob, and Charlie have to buy tickets.

The tickets cost different amounts for different age groups. Customers aged 1-17 are considered kids and cost \$10. Customers aged 18-59 are considered adults and cost \$20. Customers aged 60 or older are considered seniors and cost \$15.

Alice is A years old, Bob is B years old, and Charlie is C years old. They all need to buy a ticket. How much will the tickets cost the them altogether?

Constraints:

- $1 \leq A, B, C \leq 100$

Input:

A B C

Output:

Output a single line with the answer.

Sample Input:

```
66 12 44
```

Sample Output:

```
45
```

Alice is over 60 and is considered a senior for \$15. Bob is between the ages of 1 and 17 and is considered a child for \$10. Charlie is between the ages of 18 and 59 and is considered an adult for \$20. The total is $\$15 + \$10 + \$20 = \$45$.

§2 Whack-a-mole

It is getting hot so you decide to go into mBIT Land's arcade and play Whack-a-Mole!

The game has N mole-holes and initially, they all have a mole sticking out. The game lasts Q seconds. In second i, one of two things happen: If the mole in hole X_i is currently sticking out, you hit it with your hammer and the mole gets knocked down into its hole. If the mole in hole X_i is currently knocked down, the machine raises it back up so it is sticking out again.

The game ends the moment every mole is knocked down. Find if the game ends, and if so print when.

Constraints:

- $1 \leq N \leq 1000$
- $1 \leq Q \leq 5000$
- $1 \leq X_i \leq N \ (1 \leq i \leq Q)$

Input:

N Q
$X_1 \ \ldots \ X_Q$

Output:

If the game ends, print the second in which it does. If the game doesn't end, print -1.

Sample Input:

```
4 10
1 3 1 2 4 1 2 4 4 2
```

Sample Output:

```
6
```

After the first two seconds, moles 1 and 3 are knocked down. On second 3, mole 1 is put back up. After second 6, they are all knocked down and the game is over. Note that the array X can continue after the game ends.

§3 Ferris Wheel

The night is getting late you go for a ride on mBIT Land's beautiful Ferris Wheel.

The wheel spins at a rate of one revolution every K seconds. At second 0, you board the ride at the bottom of the wheel. Then, the wheel starts spinning clockwise. At N times throughout the ride, the wheel pauses for a moment while someone gets on at the bottom. The ith pause comes after the wheel had been spinning for T_i seconds.

During each pause, your capsule hovers where it is. You don't like stopping high above the ground so you want to prepare yourself for the pause where you will be left dangling the highest. Report which pause leaves you the highest off the ground. If there are multiple answers, print any.

Constraints:

- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq K \leq 10^9$
- $1 \leq T_i \leq 10^9$
- $T_i < T_{i+1}$ $(1 \leq i \leq N-1)$

Input:

N K
T_1 $T_2 \ldots T_N$

Output:

Output a single line with the answer. If there are multiple answers, print any.

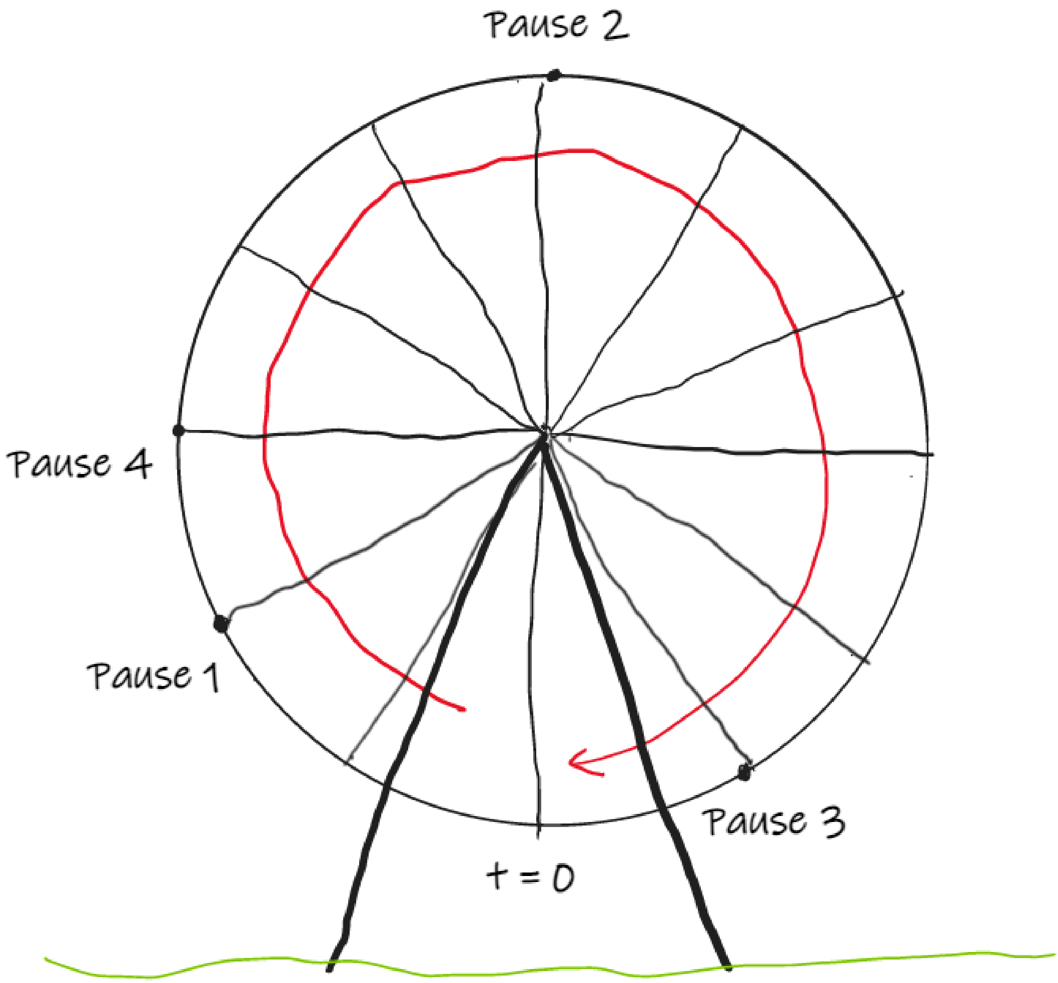
Sample Input:

```
4 12
2 6 11 15
```

Sample Output:

```
2
```

Pause 2 leaves you the highest. (See diagram on next page)


Pause 2
Pause 4
Pause 1
Pause 3
t = 0

§4 Behind the Scenes

You are at a one of mBIT Land's Behind-the-Scenes event and you're given one of the actor's scripts. Unfortunately, it's dark outside, and you can only decipher the lengths of the N words on the script. The lengths of the words are given in an array L.

You get bored if the script has the same word over and over. Therefore, you wish that the script many distinct words as possible. Print a possible list of words that could be the script, where the length of the ith word is L_i.

Note: A word is any string of letters from 'a' to 'z'. Two words are distinct if they are of different lengths or differ in at least one position.

Constraints:

- $1 \leq N \leq 1000$
- $1 \leq L_i \leq 1000$ $(1 \leq i \leq N)$
- $\sum_{i=1}^{n} L_i \leq 5000$

Input:

N
$L_1 \; \ldots \; L_N$

Output:

Output N strings, separated by space or line, satisfying the length constraint.

Sample Input:

```
5
2 3 4 2 6
```

Sample Output:

```
do you like my script
```

§5 Shifting Seats

You and your group of N friends have just boarded one of mBIT Land's coasters. Unfortunately, you didn't coordinate where you were going to sit, so you ended up sitting haphazardly in arbitrary locations.

The friends are labeled in increasing order of seat number and friend i sat down in seat A_i. Now you have decided you want to move so that you can be together in consecutive seats (for social distancing reasons) on the coaster. In one second, a friend can move up one seat (increase their seat number by 1). Multiple friends can move in the same time. There are an infinite number of seats and friends can temporarily be sitting in the same seat. What is the minimum total time needed so you can all sit in consecutive seats?

Constraints:

- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq A_i \leq 10^9$ $(1 \leq i \leq N)$
- $A_i \leq A_{i+1}$ $(1 \leq i \leq N-1)$

Input:

N
A_1 ... A_N

Output:

Print one line with the answer.

Sample Input:

```
7
2 3 3 5 7 7 7
```

Sample Output:

```
2
```

In second 1, friends 1, 2, 3, 4, 6, and 7 should move up so the positions are 3, 4, 4, 6, 7, 8, 8. In second 2, friends 3 and 7 should move up so the positions are 3, 4, 5, 6, 7, 8, 9.

§6 Roller Coaster

You are considering riding the Claire-a-Coaster, the scariest ride in all of mBIT Land.

The tracks of the Claire-a-Coaster be described by a sequence of N 2D coordinates (X_i, Y_i). Consecutive coordinates are a distance of 1 from each other in a cardinal direction. Connecting consecutive coordinates gives the path that the coaster follows.

The train itself always stays on one side of the track. The first segment of the track is guaranteed to be horizontal, and the train starts on top of this segment. Additionally, it is guaranteed all angles between three consecutive points are right angles.

You are only willing to ride the infamous Claire-a-coaster if the train never goes upside down. For T test cases, report whether it ever goes upside down.

Constraints:

- $1 \leq T \leq 1000$
- The sum of N over all test cases does not exceed $2 \cdot 10^5$

For each test case:

- $2 \leq N \leq 2 \cdot 10^5$
- $1 \leq X_i, Y_i \leq 10^9$
- $(X_{i+1}, Y_{i+1}) \in \{(X_i + 1, Y_i), (X_i - 1, Y_i), (X_i, Y_i + 1), (X_i, Y_i - 1)\}$ $(1 \leq i \leq N - 1)$
- $(X_i, Y_i) \neq (X_{i+2}, Y_{i+2})$ $(1 \leq i \leq N - 2)$
- $(X_2, Y_2) \in \{(X_1 + 1, Y_1), (X_1 - 1, Y_1)\}$

Input:

The first line contains T. Then T test cases follow. Each test case is in the form:

N
X_1 Y_1
$\vdots$
X_N Y_N

Output:

Print T lines. The ith line should contain the string `Yes` if the roller coaster goes upside down at some point and `No` if it doesn't.

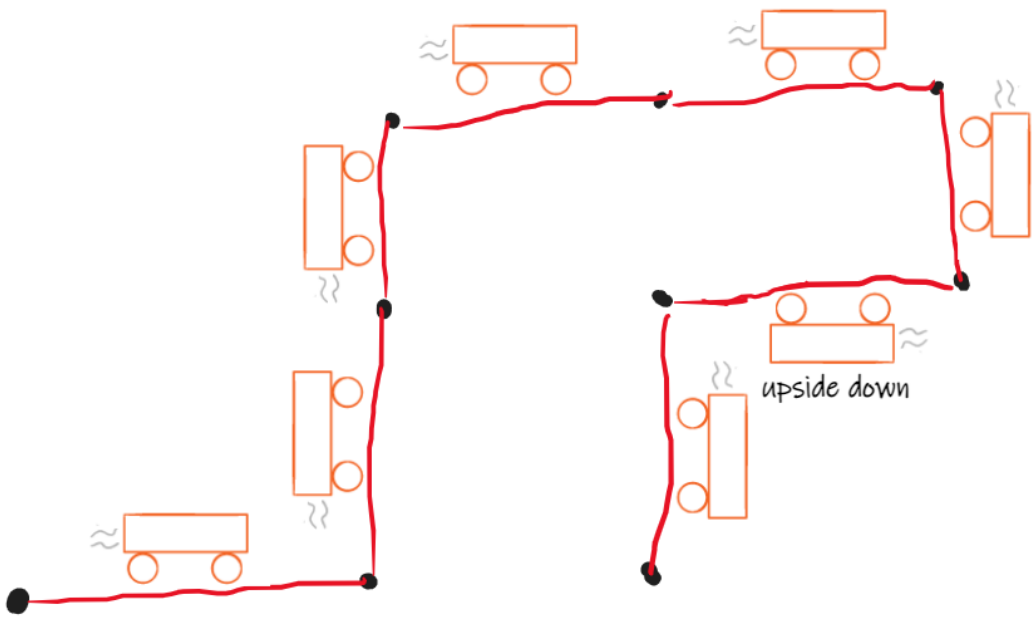
Sample Input:

```
2
9
1 1
2 1
2 2
2 3
3 3
4 3
4 2
3 2
3 1
4
3 1
2 1
2 2
1 2
```

Sample Output:

```
Yes
No
```

The first test case looks like



§7 Toadstools

You go to the park's playground and decide to jump around on the toadstools.

There are N toadstools in a line and the height of the ith one is Y_i. Toadstool i can "see" toadstool j if someone of negligible height standing on toadstool i can see the top of toadstool j without any toadstools in between blocking their line of sight. Formally, toadstool i can "see" toadstool j ($i \neq j$) if and only if the line segment from coordinate (i, Y_i) to (j, Y_j) does not intersect or even graze line segment from $(k, 0)$ to (k, Y_k) for all $i < k < j$ (see diagram after Sample Output).

Feeling bored, you wonder: how many pairs of toadstools can "see" each other?

Constraints:

- $1 \leq N \leq 1000$
- $1 \leq A_i \leq 10^9 (1 \leq i \leq N)$

Input:

N
$Y_1 \ \ldots \ Y_N$

Output:

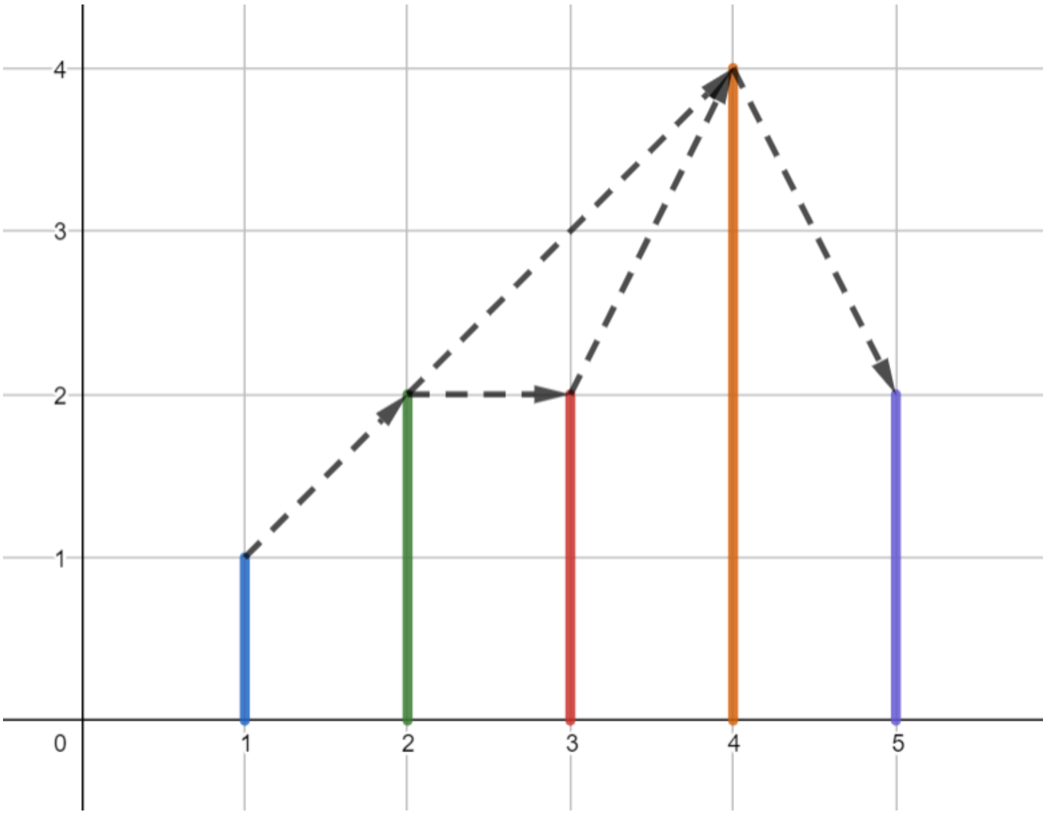
Output one line with the answer.

Sample Input:

```
5
1 2 2 4 2
```

Sample Output:

```
5
```



In the sample, the pairs of toadstools that can see each other are $(1, 2)$, $(2, 3)$, $(2, 4)$, $(3, 4)$, and $(4, 5)$. Toadstools 1 and 4 can not see each other because toadstool 2 grazes the line of sight.

§8 Snapshot

You are operating the security cameras at mBIT Land and have been alerted of possible tampering with the security photos!

There were N visitors walking around mBIT Land today. Each visitor was either walking right or left at a constant velocity of either 1 or -1, but you don't know which. You do know where each visitors started the day, represented by an increasing list X of length N where X_i is the initial location of visitor i (at time $t = 0$).

The only other information you have is K time-stamped photos. The ith was taken at time T_i and is described by L_i, a non-decreasing list of length N where $L_{i,j}$ is the location of a visitor in photo i. In other words, each picture consists of the N visitors' collective locations at a known time.

Unfortunately, your cameras are blurry and you can't decipher who is who across the photos. Can you deduce whether the photos have been tampered with? If the photos are consistent find the initial velocity of each visitor. If one of them has been tampered with, report it.

Constraints:

- $1 \le N \cdot K \le 2 \cdot 10^5$
- $1 \le T_i \le 10^9$
- $T_i < T_{i+1}$ $(1 \le i \le K-1)$
- $-10^9 \le X_i \le 10^9$ $(1 \le i \le N)$
- $X_i < X_{i+1}$ $(1 \le i \le N-1)$
- $-2 \cdot 10^9 \le L_{i,j} \le 2 \cdot 10^9$ $(1 \le i \le K,\ 1 \le j \le N)$
- $L_{i,j} \le L_{i,j+1}$ $(1 \le i \le K,\ 1 \le j \le N-1)$

Input:

$N\ K$
$X_1\ \ldots\ X_N$
T_1
$L_{1,1}\ \ldots L_{1,N}$
$\vdots$
T_K
$L_{K,1}\ \ldots L_{K,N}$

Output:

If the photos have been tampered with, print

0

Otherwise, print the velocities of the visitors

$V_1\ \ldots\ V_N$

Sample Input:

```
5 3
1 3 6 7 11
2
-1 4 5 5 13
5
-4 1 2 8 16
7
-6 -1 0 10 18
```

Sample Output:

```
-1 1 -1 -1 1
```

§9 Park Passes

You are obsessed with mBIT Land and want to go on rides every day!

There are N ticket packages you can buy. Package i consists of T_i tickets, expires after day E_i, and costs C_i dollars. When a ticket package expires, every ticket that came with the package can no longer be used. You want to have an unexpired ticket to use for every day from day 1 to day M.

What is the minimum amount you need to spend to achieve your goal? If there is no way to buy packages such that you can go every day, report it.

Constraints:

- $1 \leq N \leq 1000$
- $1 \leq M \leq 1000$
- $1 \leq T_i \leq E_i \leq M$
- $1 \leq C_i \leq 10^9$

Input:

$N\ M$
$T_1\ E_1\ C_1$
$\vdots$
$T_N\ E_N\ C_N$

Output:

If it is impossible to go every day, print `-1`. Otherwise, print the answer.

Sample Input:

```
4 6
1 6 5
4 4 3
3 5 2
2 2 2
```

Sample Output:

```
9
```

It is optimal to buy packages 1, 3, and 4.

§10 Street Performers

You have access to the schedule of street performances on Blair Boulevard and want to watch all of them.

The performers are labeled 1 through N in the order that they perform in. Performance i will take place at location A_i on the street represented by a number line.

You want to be able to sit down on your lawn chair while you watch the performances. From location X, you can see any performance between locations X and $X+K$, inclusive. You don't like having to move your setup, but are sometimes forced to when the next performance is unviewable from your current location.

Unfortunately, the weather report has told you that some of the performances at the beginning might be rained out. You want to be prepared for any number of of the performances at the beginning to be cancelled. For every suffix of performances, find the minimum number of times you will have to move to catch every performance if you can freely choose your starting location and where you move to each time.

Constraints:

- $1 \leq N \leq 2 \cdot 10^5$
- $1 \leq K \leq 10^9$
- $1 \leq A_i \leq 10^9$ $(1 \leq i \leq N)$

Input:

N K
A_1 ... A_N

Output:

Print a line with N integers. The ith integer should be the answer for the suffix starting at performance i.

Sample Input:

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

Sample Output:

```
3 3 2 2 2 1 1 0
```

For the suffix starting at 1 (all performances), an optimal solution is to first set up at location 4 to see performances 1-3 that are all in the range $[4, 4+4]$. You can't see performance 2 from location 4 so you move to location 1 where you can see performances 4 and 5. Then, move to location 2 to see performances 6 and 7. Finally move to location 7 to see the last performance.

§11 Rating Report

The National Amusement Park Society (NAPS) is tasked with submitting mBIT Land's annual rating report. The committee consists of N amusement-park-reviewers who will each submit their rating of mBIT Land.

The reviewers work in a hierarchy in which each reviewer, except reviewer 1 has a supervisor to whom they report to. Reviewer i's supervisor is P_i $(2 \leq i \leq N)$. Out of the N reviewers, K of them have no subordinates (no one whom they are the supervisor to). Each of these K reviewers will generate a first-hand rating of mBIT Land by visiting the park. Some of them have already done this, but some of them have yet to formulate their opinion, meaning you can still influence their rating. You can influence the undecided first-hand reviewers' ratings to anything so long as all K first-hand ratings are in the range $[1, K]$ and pairwise distinct.

The other $N - K$ reviewers are lazy and use their subordinate's ratings to generate their own ratings. Each of these $N - K$ reviewers is either an optimist, in which case their rating is equal to the best rating out their subordinates, or a pessimist, in which case their rating is equal to the worst rating out of their subordinates. After generating their rating, each reviewer reports their rating to their supervisor. No reviewer generates their rating until each of their subordinates has generated their rating.

As input, you're given an array A:

- $A_i = 0$ means that i is a first-hand reviewer who hasn't submitted a rating yet
- $A_i > 0$ means i is a first-hand reviewer who has already submitted a rating $R_i = A_i$
- $A_i = \texttt{max}$ means that i is not a first-hand reviewer, and is an optimist who submits a rating $R_i := \max\limits_{j|P_j=i} R_j$ (j is a subordinate of i)
- $A_i = \texttt{min}$ means that i is not a first-hand reviewer, and is a pessimist who submits a rating $R_i := \min\limits_{j|P_j=i} R_j$ (j is a subordinate of i)

Ultimately, the manager (reviewer 1) submits R_1 as the park's final rating. As an avid supporter of mBIT Land, find the maximum possible final rating you can achieve by influencing the first-hand reviewers optimally.

Constraints:

- $2 \leq N \leq 2 \cdot 10^5$
- $1 \leq K \leq N$
- $1 \leq P_i \leq i - 1$ $(2 \leq i \leq N)$
- The tree formed has K leaves (first-hand reviewers)
- A_i is an integer between 0 and K if i is a leaf
- $A_i \neq A_j$ if i is a leaf, j is a leaf, $A_i > 0$, and $A_j > 0$
- A_i is one of $\{\texttt{max}, \texttt{min}\}$ if i is not a leaf

Input:

You are given N, K, the array P, and the array A.

N K
P_2 ... P_N
A_1
$\vdots$
A_N

Output:

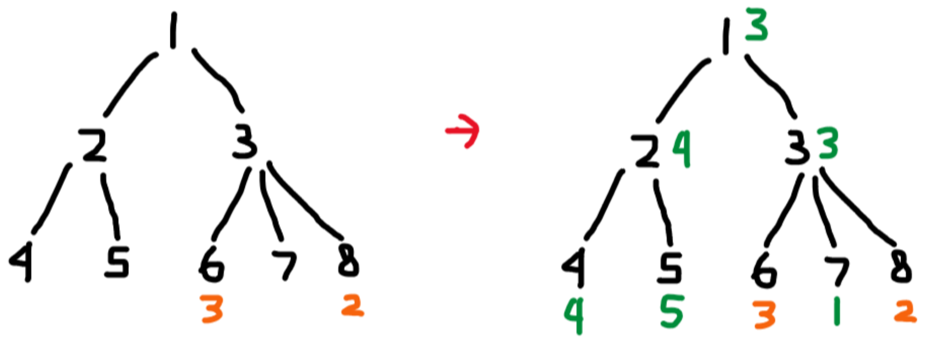
Output one line with maximum possible R_1.

Sample Input:

```
8 5
1 1 2 2 3 3 3
min
min
max
0
0
3
0
2
```

Sample Output:

```
3
```



One optimal assignment of first-hand ratings is:

$$R_4 = 4$$
$$R_5 = 5$$
$$R_7 = 1$$

§12 Sugar Rush

You are addicted to cotton candy!

There are N cotton candy stands along Blair Boulevard. The ith stand is located at X_i meters along the park. No two stands are at the same location.

You start by visiting one of the cotton candy stands. After the visiting a stand, you recalculate which unvisited stand is closest to your current position and rush to that stand. If there are multiple closest stands, you go to the one with the highest coordinate.

Being indecisive, you don't know which stand you should visit first. For each $i = 1, \dots, N$, find the total distance you will travel to visit every cotton candy stand if you started at stand i.

Constraints:

- $2 \leq N \leq 2 \cdot 10^5$
- $1 \leq X_i \leq 10^9$ $(1 \leq i \leq N)$
- $X_i < X_{i+1}$ $(1 \leq i \leq N-1)$

Input:

N
$X_1 \ \dots \ X_N$

Output:

Output N lines with the respective answers.

Sample Input:

```
7
2 5 6 10 12 20 22
```

Sample Output:

```
20
37
24
32
30
22
20
```

§13 Not Nim

Your nerdy competitive programming friend is tagging along throughout your trip in mBIT Land. Your friend doesn't enjoy anything at the park and instead wants to play a game with you.

> There's N piles of stones, the ith having P_i stones. Two players alternate taking stones -

You cannot let your friend try to entice you with another combinatorial game. Your friend senses your irritation and assures you this is *Not Nim*.

> In *Not Nim*, two players alternate taking a *fixed* number of stones from any pile. Let the two players be Alice and Bob, with Alice going first. On Alice's turn, she takes exactly A stones from a pile of her choice, or loses if there is no pile with at least A stones. On Bob's turn, he takes exactly B stones from a pile of his choice, or loses if there is no pile with at least B stones.

Seeing your sustained apathy, your friend makes a compromise: if you can predict the winner for T games, you don't have to play.

So, who wins each game, assuming Alice and Bob always play optimally?

Constraints:

- $1 \leq T \leq 2 \cdot 10^5$
- $1 \leq \sum N \leq 2 \cdot 10^5$
- $1 \leq A, B \leq 10^9$
- $1 \leq P_i \leq 10^9$ $(1 \leq i \leq N)$

Input:

The first line is T, the number of games. Each game consists of the following input:

N
A B
P_1 ... P_N

Output:

Output T lines. On the ith print the winner of the ith game, either `Alice` or `Bob`.

Sample Input:

```
2
3
3 2
2 4 4
4
4 4
7 2 9 12
```

Sample Output:

```
Bob
Bob
```

In the first game, Alice must take from a pile of 4 on her first turn. Then, Bob can take 2 from the other pile of 4 and Alice will be unable to move.

§14 Mascot Parade

The mascots of mBIT Land are having a parade! They are all in vehicles and are traveling down Blair Boulevard, waving and blowing kisses to happy children.

There N mascots are numbered 1 through N in increasing order of position. The ith $(1 \leq i \leq N)$ mascot has a speed S_i and an initial position P_i. This means that every second they have the potential to advance S_i meters, and at second 0 they are at position P_i meters down the road.

The road is very narrow, which means the mascots can't pass each other. Instead, if i is travelling faster than $i+1$, then on the second in which mascot i's position would be greater than or equal to mascot $i+1$'s position, mascot i instead gets held back and will lag exactly one meter behind mascot $i+1$ and travel at the same speed of mascot $i+1$ for the rest of the parade.

The children can only come to the parade at certain times and only want to see a certain mascot. There are Q children and child i wants to know how far down the road mascot X_i will be at time Y_i. You have to answer the children's queries in order or else they will get upset (see Input for details).

Help the parade run smoothly by telling the children where their favorite mascots will be!

Constraints:

- $1 \leq N, Q \leq 2 \cdot 10^5$
- $1 \leq P_i, S_i \leq 10^9$ $(1 \leq i \leq N)$
- $P_i < P_{i+1}$ $(1 \leq i \leq N-1)$
- $0 \leq A_i \leq N-1$ $(1 \leq i \leq Q)$
- $0 \leq B_i \leq 10^9 - 1$ $(1 \leq i \leq Q)$

Input:

N Q
$P_1 \ldots P_N$
$S_1 \ldots S_N$
A_1 B_1
$\vdots$
A_Q B_Q

Let ans be the answer to the previous query or 0 if there are no previous queries. The ith query $(1 \leq i \leq N)$ is decoded as follows:

$$X_i = (ans + A_i)\%N + 1$$
$$Y_i = (ans + B_i)\%10^9 + 1$$

Output:

Output Q lines with the answers to the queries.

Sample Input:

```
7 4
3 4 7 9 11 12 20
2 5 1 8 7 10 4
0 3
1 999999992
2 999999976
2 999999981
```

Sample Output:

```
9
24
22
33
```

The input gets decoded to the following queries:

```
1 4
4 2
6 1
4 4
```

The following table shows the initial locations and speeds of the mascots.

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$
S_i	2	5	1	8	7	10	4
P_i	3	4	7	9	11	12	20

Below are their positions in the first 4 seconds:

$t = 1$	5	7	8	17	18	22	24
$t = 2$	7	8	9	24	25	27	28
$t = 3$	8	9	10	29	30	31	32
$t = 4$	9	10	11	33	34	35	36