A. Reposts

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Polycarp published a funny picture in a social network making a poll about the color of his handle. Many of his friends started reposting Polycarp's joke to their news feed. Some of them reposted the reposts and so on.

These events are given as a sequence of strings "*name1* `reposted` *name2*", where *name1* is the name of the person who reposted the joke, and *name2* is the name of the person from whose news feed the joke was reposted. It is guaranteed that for each string "*name1* `reposted` *name2*" user "*name1*" didn't have the joke in his feed yet, and "*name2*" already had it in his feed by the moment of repost. Polycarp was registered as "`Polycarp`" and initially the joke was only in his feed.

Polycarp measures the popularity of the joke as the length of the largest repost chain. Print the popularity of Polycarp's joke.

Input

The first line of the input contains integer n ($1 \le n \le 200$) — the number of reposts. Next follow the reposts in the order they were made. Each of them is written on a single line and looks as "*name1* `reposted` *name2*". All the names in the input consist of lowercase or uppercase English letters and/or digits and have lengths from 2 to 24 characters, inclusive.

We know that the user names are case-insensitive, that is, two names that only differ in the letter case correspond to the same social network user.

Output

Print a single integer — the maximum length of a repost chain.

Examples

input
5 tourist reposted Polycarp Petr reposted Tourist WJMZBMR reposted Petr sdya reposted wjmzbmr vepifanov reposted sdya
output
6

input
6 Mike reposted Polycarp Max reposted Polycarp EveryOne reposted Polycarp 111 reposted Polycarp VkCup reposted Polycarp Codeforces reposted Polycarp
output
2

input
1 SoMeStRaNgEgUe reposted PoLyCaRp
output
2

B. Photo to Remember

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day n friends met at a party, they hadn't seen each other for a long time and so they decided to make a group photo together.

Simply speaking, the process of taking photos can be described as follows. On the photo, each photographed friend occupies a rectangle of pixels: the i-th of them occupies the rectangle of width w_i pixels and height h_i pixels. On the group photo everybody stands in a line, thus the minimum pixel size of the photo including all the photographed friends, is $W \times H$, where W is the total sum of all widths and H is the maximum height of all the photographed friends.

As is usually the case, the friends made n photos — the j-th ($1 \le j \le n$) photo had everybody except for the j-th friend as he was the photographer.

Print the minimum size of each made photo in pixels.

Input

The first line contains integer n ($2 \le n \le 200\,000$) — the number of friends.

Then n lines follow: the i-th line contains information about the i-th friend. The line contains a pair of integers w_i, h_i ($1 \le w_i \le 10, 1 \le h_i \le 1000$) — the width and height in pixels of the corresponding rectangle.

Output

Print n space-separated numbers $b_1, b_2, \ldots, b_n$, where b_i — the total number of pixels on the minimum photo containing all friends expect for the i-th one.

Examples

input
3 1 10 5 5 10 1
output
75 110 60

input
3 2 1 1 2 2 1
output
6 4 6

C. Chicken or Fish?

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polycarp is flying in the airplane. Finally, it is his favorite time — the *lunchtime*. The BerAvia company stewardess is giving food consecutively to all the passengers from the 1-th one to the last one. Polycarp is sitting on seat m, that means, he will be the m-th person to get food.

The flight menu has k dishes in total and when Polycarp boarded the flight, he had time to count the number of portions of each dish on board. Thus, he knows values $a_1, a_2, \ldots, a_k$, where a_i is the number of portions of the i-th dish.

The stewardess has already given food to $m - 1$ passengers, gave Polycarp a polite smile and asked him what he would prefer. That's when Polycarp realized that they might have run out of some dishes by that moment. For some of the $m - 1$ passengers ahead of him, he noticed what dishes they were given. Besides, he's heard some strange mumbling from some of the $m - 1$ passengers ahead of him, similar to phrase 'I'm disappointed'. That happened when a passenger asked for some dish but the stewardess gave him a polite smile and said that they had run out of that dish. In that case the passenger needed to choose some other dish that was available. If Polycarp heard no more sounds from a passenger, that meant that the passenger chose his dish at the first try.

Help Polycarp to find out for each dish: whether they could have run out of the dish by the moment Polyarp was served or that dish was definitely available.

Input

Each test in this problem consists of one or more input sets. First goes a string that contains a single integer t ($1 \le t \le 100\,000$) — the number of input data sets in the test. Then the sets follow, each set is preceded by an empty line.

The first line of each set of the input contains integers m, k ($2 \le m \le 100\,000, 1 \le k \le 100\,000$) — the number of Polycarp's seat and the number of dishes, respectively.

The second line contains a sequence of k integers $a_1, a_2, \ldots, a_k$ ($1 \le a_i \le 100\,000$), where a_i is the initial number of portions of the i-th dish.

Then $m - 1$ lines follow, each line contains the description of Polycarp's observations about giving food to a passenger sitting in front of him: the j-th line contains a pair of integers t_j, r_j ($0 \le t_j \le k, 0 \le r_j \le 1$), where t_j is the number of the dish that was given to the j-th passenger (or 0, if Polycarp didn't notice what dish was given to the passenger), and r_j — a 1 or a 0, depending on whether the j-th passenger was or wasn't disappointed, respectively.

We know that sum a_i equals at least m, that is,Polycarp will definitely get some dish, even if it is the last thing he wanted. It is guaranteed that the data is consistent.

Sum m for all input sets doesn't exceed $100\,000$. Sum k for all input sets doesn't exceed $100\,000$.

Output

For each input set print the answer as a single line. Print a string of k letters "Y" or "N". Letter "Y" in position i should be printed if they could have run out of the i-th dish by the time the stewardess started serving Polycarp.

Examples

input
2 3 4 2 3 2 1 1 0 0 0 5 5 1 2 1 3 1 3 0 0 0 2 1 4 0

output
YNNY YYYNY

Note

In the first input set depending on the choice of the second passenger the situation could develop in different ways:

- If he chose the first dish, then by the moment the stewardess reaches Polycarp, they will have run out of the first dish;
- If he chose the fourth dish, then by the moment the stewardess reaches Polycarp, they will have run out of the fourth dish;

- Otherwise, Polycarp will be able to choose from any of the four dishes.

Thus, the answer is "YNNY".

In the second input set there is, for example, the following possible scenario. First, the first passenger takes the only third dish, then the second passenger takes the second dish. Then, the third passenger asks for the third dish, but it is not available, so he makes disappointed muttering and ends up with the second dish. Then the fourth passenger takes the fourth dish, and Polycarp ends up with the choice between the first, fourth and fifth dish.

Likewise, another possible scenario is when by the time the stewardess comes to Polycarp, they will have run out of either the first or the fifth dish (this can happen if one of these dishes is taken by the second passenger). It is easy to see that there is more than enough of the fourth dish, so Polycarp can always count on it. Thus, the answer is "YYYNY".

D. Closest Equals

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given sequence $a_1, a_2, \ldots, a_n$ and m queries l_j, r_j ($1 \le l_j \le r_j \le n$). For each query you need to print the minimum distance between such pair of elements a_x and a_y ($x \ne y$), that:

- both indexes of the elements lie within range $[l_j, r_j]$, that is, $l_j \le x, y \le r_j$;
- the values of the elements are equal, that is $a_x = a_y$.

The text above understands distance as $|x - y|$.

Input

The first line of the input contains a pair of integers n, m ($1 \le n, m \le 5 \cdot 10^5$) — the length of the sequence and the number of queries, correspondingly.

The second line contains the sequence of integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$).

Next m lines contain the queries, one per line. Each query is given by a pair of numbers l_j, r_j ($1 \le l_j \le r_j \le n$) — the indexes of the query range limits.

Output

Print m integers — the answers to each query. If there is no valid match for some query, please print -1 as an answer to this query.

Examples

input
5 3 1 1 2 3 2 1 5 2 4 3 5
output
1 -1 2

input
6 5 1 2 1 3 2 3 4 6 1 3 2 5 2 4 1 6
output
2 2 3 -1 2