Codeforces Round 966 (Div. 3)

A. Primary Task

1 second, 256 megabytes

Dmitry wrote down t integers on the board, and that is good. He is sure that he lost an *important* integer n among them, and that is bad.

The integer n had the form $10\hat{}x$ ($x \geq 2$), where the symbol '^' denotes exponentiation.. Something went wrong, and Dmitry missed the symbol '^' when writing the *important* integer. For example, instead of the integer 10^5, he would have written 105, and instead of 10^{19}, he would have written 1019.

Dmitry wants to understand which of the integers on the board could have been the *important* integer and which could not.

Input

The first line of the input contains one integer t ($1 \leq t \leq 10^4$) — the number of integers on the board.

The next t lines each contain an integer a ($1 \leq a \leq 10000$) — the next integer from the board.

Output

For each integer on the board, output "YES" if it could have been the *important* integer and "NO" otherwise.

You may output each letter in any case (lowercase or uppercase). For example, the strings "yEs", "yes", "Yes", and "YES" will be accepted as a positive answer.

input
7 100 1010 101 105 2033 1019 1002

output
NO YES NO YES NO YES NO

B. Seating in a Bus

2 seconds, 256 megabytes

In Berland, a bus consists of a row of n seats numbered from 1 to n. Passengers are advised to always board the bus following these rules:

- If there are no occupied seats in the bus, a passenger can sit in any free seat;
- Otherwise, a passenger should sit in any free seat that has at least one occupied neighboring seat. In other words, a passenger can sit in a seat with index i ($1 \leq i \leq n$) only if at least one of the seats with indices $i - 1$ or $i + 1$ is occupied.

Today, n passengers boarded the bus. The array a chronologically records the seat numbers they occupied. That is, a_1 contains the seat number where the first passenger sat, a_2 — the seat number where the second passenger sat, and so on.

You know the contents of the array a. Determine whether all passengers followed the recommendations.

For example, if $n = 5$, and a = [$5, 4, 2, 1, 3$], then the recommendations were not followed, as the 3-rd passenger sat in seat number 2, while the neighboring seats with numbers 1 and 3 were free.

Input

The first line of input contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The following describes the input test cases.

The first line of each test case contains exactly one integer n ($1 \leq n \leq 2 \cdot 10^5$) — the number of seats in the bus and the number of passengers who boarded the bus.

The second line of each test case contains n **distinct** integers a_i ($1 \leq a_i \leq n$) — the seats that the passengers occupied in chronological order.

It is guaranteed that the sum of n values across all test cases does not exceed $2 \cdot 10^5$, and that no passenger sits in an already occupied seat.

Output

For each test case, output on a separate line:

- "YES", if all passengers followed the recommendations;
- "NO" otherwise.

You may output the answer in any case (for example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as a positive answer).

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

output
NO YES YES NO

The first test case is explained in the problem statement.

C. Numeric String Template

2 seconds, 256 megabytes

Kristina has an array a, called a *template*, consisting of n integers. She also has m strings, each consisting only of lowercase Latin letters. The strings are numbered from 1 to m. She wants to check which strings match the template.

A string s is considered to match the template if all of the following conditions are simultaneously satisfied:

- The length of the string s is equal to the number of elements in the array a.
- The same numbers from a correspond to the same symbols from s. So, if $a_i = a_j$, then $s_i = s_j$ for ($1 \leq i, j \leq n$).
- The same symbols from s correspond to the same numbers from a. So, if $s_i = s_j$, then $a_i = a_j$ for ($1 \leq i, j \leq n$).

In other words, there must be a one-to-one correspondence between the characters of the string and the elements of the array.
For example, if a = [$3, 5, 2, 1, 3$], then the string "abfda" matches the template, while the string "afbfa" does not, since the character "f" corresponds to both numbers 1 and 5.

Input

The first line of input contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

The following descriptions are for the test cases.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of elements in the array a.

The second line of each test case contains exactly n integers a_i ($-10^9 \le a_i \le 10^9$) — the elements of the array a.

The third line of each test case contains a single integer m ($1 \le m \le 2 \cdot 10^5$) — the number of strings to check for template matching.

Following are m strings, each containing a non-empty string s_j ($1 \le |s_j| \le 2 \cdot 10^5$), consisting of lowercase Latin letters.

It is guaranteed that the sum of n across all test cases does not exceed $2 \cdot 10^5$, and that the sum of the lengths of all strings does not exceed $2 \cdot 10^5$.

Output

For each test case, output m lines. On the i-th line ($1 \le i \le m$) output:

- "YES", if the string with index i matches the template;
- "NO" otherwise.

You may output the answer in any case (for example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as a positive answer).

input
3 5 3 5 2 1 3 2 abfda afbfa 2 1 2 3 ab abc aa 4 5 -3 5 -3 4 aaaa bcbc aba cbcb

output
YES NO YES NO NO NO YES NO YES

The first test case is explained in the problem statement.

D. Right Left Wrong

2 seconds, 256 megabytes

Vlad found a strip of n cells, numbered from left to right from 1 to n. In the i-th cell, there is a positive integer a_i and a letter s_i, where all s_i are either 'L' or 'R'.

Vlad invites you to try to score the maximum possible points by performing any (possibly zero) number of operations.

In one operation, you can choose two indices l and r ($1 \le l < r \le n$) such that s_l = 'L' and s_r = 'R' and do the following:

- add $a_l + a_{l+1} + \cdots + a_{r-1} + a_r$ points to the current score;
- replace s_i with '.' for all $l \le i \le r$, meaning you can no longer choose these indices.

For example, consider the following strip:

3	5	1	4	3	2
L	R	L	L	L	R

You can first choose $l = 1$, $r = 2$ and add $3 + 5 = 8$ to your score.

3	5	1	4	3	2
.	.	L	L	L	R

Then choose $l = 3$, $r = 6$ and add $1 + 4 + 3 + 2 = 10$ to your score.

3	5	1	4	3	2
.	.	.	.	.	.

As a result, it is impossible to perform another operation, and the final score is 18.

What is the maximum score that can be achieved?

Input

The first line contains one integer t ($1 \le t \le 10^4$) — the number of test cases.

The first line of each test case contains one integer n ($2 \le n \le 2 \cdot 10^5$) — the length of the strip.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^5$) — the numbers written on the strip.

The third line of each test case contains a string s of n characters 'L' and 'R'.

It is guaranteed that the sum of the values of n across all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output one integer — the maximum possible number of points that can be scored.

input
4 6 3 5 1 4 3 2 LRLLLR 2 2 8 LR 2 3 9 RL 5 1 2 3 4 5 LRLRR

output
18 10 0 22

E. Photoshoot for Gorillas

2 seconds, 256 megabytes

You really love gorillas, so you decided to organize a photoshoot for them. Gorillas live in the jungle. The jungle is represented as a grid of n rows and m columns. w gorillas agreed to participate in the photoshoot, and the gorilla with index i ($1 \le i \le w$) has a *height* of a_i. You want to place **all** the gorillas in the cells of the grid such that there is **no more than one gorilla** in each cell.

The *spectacle* of the arrangement is equal to the sum of the *spectacles* of all sub-squares of the grid with a side length of k.

The *spectacle* of a sub-square is equal to the sum of the *heights* of the gorillas in it.

From all suitable arrangements, choose the arrangement with the **maximum** *spectacle*.

Input

The first line contains an integer t ($1 \le t \le 10^3$) — the number of test cases.

The descriptions of the test cases follow.

The first line contains integers n, m, k ($1 \le n, m \le 2 \cdot 10^5$, $1 \le n \cdot m \le 2 \cdot 10^5$, $1 \le k \le \min(n, m)$) — the dimensions of the grid and the side length of the square.

The second line contains an integer w ($1 \le w \le n \cdot m$) — the number of gorillas.

The third line contains w integers $a_1, a_2, \ldots, a_w$ ($1 \le a_i \le 10^9$) — the *heights* of the gorillas.

It is guaranteed that the sum of $n \cdot m$ across all test cases does not exceed $2 \cdot 10^5$. The same guarantee applies to w.

Output

For each test case, output a single integer — the **maximum** *spectacle* of a suitable arrangement.

input
5 3 4 2 9 1 1 1 1 1 1 1 1 1 2 1 1 2 5 7 20 15 7 9 4 1 4 5 6 1 1000000000 898 777 1984 1 1 4 5 4 1499 2004 9 5 5 6 6 7 14 16 16 6

output
21 12 49000083104 3512 319

In the first test case of the first input set, the *spectacle* of the following sub-squares is summed:

Yellow color corresponds to the sub-squares, green — to the rest of the grid squares.

The picture shows the optimal arrangement of the gorillas. The *spectacle* of the arrangement is $4 + 4 + 3 + 3 + 4 + 3 = 21$.

F. Color Rows and Columns

3 seconds, 256 megabytes

You have n rectangles, the i-th of which has a width of a_i and a height of b_i.

You can perform the following operation an unlimited number of times: choose a rectangle and a cell in it, and then color it.

Each time you completely color any row or column, you earn 1 point. Your task is to score at least k points with as few operations as possible.

Suppose you have a rectangle with a width of 6 and a height of 3. You can score 4 points by coloring all the cells in any 4 columns, thus performing 12 operations.

Input

The first line contains an integer t ($1 \le t \le 100$) — the number of test cases. The following are the descriptions of the test cases.

The first line of each test case description contains two integers n and k ($1 \le n \le 1000, 1 \le k \le 100$) — the number of rectangles in the case and the required number of points.

The next n lines contain the descriptions of the rectangles. The i-th line contains two integers a_i and b_i ($1 \le a_i, b_i \le 100$) — the width and height of the i-th rectangle.

It is guaranteed that the sum of the values of n across all test cases does not exceed 1000.

Output

For each test case, output a single integer — the minimum number of operations required to score at least k points. If it is impossible to score at least k points, output `-1`.

input
7 1 4 6 3 1 5 4 4 5 10 1 1 1 1 1 1 1 1 2 100 1 2 5 6 3 11 2 2 3 3 4 4 3 25 9 2 4 3 8 10 4 18 5 4 8 5 8 3 6 2

output
12 14 5 -1 17 80 35

G. Call During the Journey

4 seconds, 256 megabytes

You live in a city consisting of n intersections and m streets connecting some pairs of intersections. You can travel in either direction on each street. No two streets connect the same pair of intersections, and no street connects an intersection to itself. You can reach any intersection from any other, possibly passing through some other intersections.

Every minute, you can board a bus at intersection u_i and travel for l_{i1} minutes to intersection v_i. Conversely, you can travel from intersection v_i to intersection u_i in l_{i1} minutes. You can only board and exit the bus at intersections. You can only board the bus at an intersection if you are currently there.

You can also walk along each street, which takes $l_{i2} > l_{i1}$ minutes.

You can make stops at intersections.

You live at intersection number 1. Today you woke up at time 0, and you have an important event scheduled at intersection number n, which you must reach no later than time t_0. You also have a phone call planned that will last from t_1 to t_2 minutes ($t_1 < t_2 < t_0$).

During the phone call, you cannot ride the bus, but you can walk along any streets, make stops, or stay at home. You can exit the bus at minute t_1 and board the bus again at minute t_2.

Since you want to get enough sleep, you became curious — how late can you leave home to have time to talk on the phone and still not be late for the event?

Input

The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases. The following are the descriptions of the test cases.

The first line of each test case contains two integers n, m ($2 \le n \le 10^5, 1 \le m \le 10^5$) — the number of intersections and streets in the city.

The second line of each test case contains three integers t_0, t_1, t_2 ($1 \le t_1 < t_2 < t_0 \le 10^9$) — the start time of the event, the start time of the phone call, and its end time, respectively.

The next m lines of each test case contain descriptions of the streets.

The i-th line contains four integers u_i, v_i, l_{i1}, l_{i2} ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le l_{i1} < l_{i2} \le 10^9$) — the numbers of the intersections connected by the i-th street, as well as the travel time along the street by bus and on foot. It is guaranteed that no two streets connect the same pair of intersections and that it is possible to reach any intersection from any other.

It is guaranteed that the sum of the values of n across all test cases does not exceed 10^5. It is also guaranteed that the sum of the values of m across all test cases does not exceed 10^5.

Output

For each test case, output a single integer — the latest time you can leave home to have time to talk on the phone and not be late for the event. If you cannot reach the event on time, output -1.

input

```
7
5 5
100 20 80
1 5 30 100
1 2 20 50
2 3 20 50
3 4 20 50
4 5 20 50
2 1
100 50 60
1 2 55 110
4 4
100 40 60
1 2 30 100
2 4 30 100
1 3 20 50
3 4 20 50
3 3
100 80 90
1 2 1 10
2 3 10 50
1 3 20 21
3 2
58 55 57
2 1 1 3
2 3 3 4
2 1
12 9 10
2 1 6 10
5 5
8 5 6
2 1 1 8
2 3 4 8
4 2 2 4
5 3 3 4
4 5 2 6
```

output

```
0
-1
60
80
53
3
2
```

H. Ksyusha and the Loaded Set

3 seconds, 512 megabytes

Ksyusha decided to start a game development company. To stand out among competitors and achieve success, she decided to write her own game engine. The engine must support a set initially consisting of n distinct integers $a_1, a_2, \ldots, a_n$.

The set will undergo m operations sequentially. The operations can be of the following types:

- Insert element x into the set;
- Remove element x from the set;
- Report the k-load of the set.

The k-load of the set is defined as the **minimum positive** integer d such that the integers $d, d+1, \ldots, d+(k-1)$ do not appear in this set. For example, the 3-load of the set $\{3, 4, 6, 11\}$ is 7, since the integers $7, 8, 9$ are absent from the set, and no smaller value fits.

Ksyusha is busy with management tasks, so you will have to write the engine. Implement efficient support for the described operations.

Input

The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases.

The following lines describe the test cases.

The first line contains an integer n ($1 \le n \le 2 \cdot 10^5$) — the initial size of the set.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_1 < a_2 < \ldots < a_n \le 2 \cdot 10^6$) — the initial state of the set.

The third line contains an integer m ($1 \le m \le 2 \cdot 10^5$) — the number of operations.

The next m lines contain the operations. The operations are given in the following format:

- + x ($1 \le x \le 2 \cdot 10^6$) — insert element x into the set (it is guaranteed that x is not in the set);
- - x ($1 \le x \le 2 \cdot 10^6$) — remove element x from the set (it is guaranteed that x is in the set);
- ? k ($1 \le k \le 2 \cdot 10^6$) — output the value of the k-load of the set.

It is guaranteed that the sum of n across all test cases does not exceed $2 \cdot 10^5$, and the same holds for m.

Output

For each test case, output the answers to the operations of type "?".

input
3 5 1 2 5 905 2000000 15 - 2 ? 2 ? 1 - 1 ? 1 + 4 + 2 ? 2 + 6 - 4 + 7 ? 2 ? 3 ? 4 ? 2000000 5 3 4 5 6 8 9 ? 5 - 5 ? 5 + 1 ? 2 - 6 - 8 + 6 ? 5 5 6 7 8 9 10 10 ? 5 - 6 ? 4 - 10 + 5 - 8 + 3 + 2 - 3 + 10

output
2 2 1 6 3 8 8 2000001 9 9 9 7 1 1