CodeForces Problem

June 06, 2023

A. Cipher Shifer

Constriants

Time Limit 1 seconds **Memory Limit** 256 MB

Problem Statement

There is a string a (unknown to you), consisting of lowercase Latin letters, encrypted according to the following rule into string s: after each character of string a, an arbitrary (possibly zero) number of any lowercase Latin letters, different from the character itself, is added; after each such addition, the character that we supplemented is added.

You are given string s, and you need to output the initial string a. In other words, you need to decrypt string s.

Note that each string encrypted in this way is decrypted uniquely.

Input Description

The first line of the input contains a single integer t ($1 \leq t \leq 1000$) — the number of test cases.

The descriptions of the test cases follow.

The first line of each test case contains a single integer n ($2 \leq n \leq 100$) — the length of the encrypted message.

The second line of each test case contains a string s of length n — the encrypted message obtained from some string a.

Output Description :

For each test case, output the decrypted message a on a separate line.

Examples

Input
3
8
abacabac
5
qzxcq
20
ccooddeeffoorrcceess
Output
ac
q
codeforces

Note

In the first encrypted message, the letter a is encrypted as aba, and the letter c is encrypted as $cabac$.
In the second encrypted message, only one letter q is encrypted as $qzxcq$.
In the third encrypted message, zero characters are added to each letter.

B. Binary Cafe

Constriants

Time Limit 1 seconds **Memory Limit** 256 MB

Problem Statement

Once upon a time, Toma found himself in a binary cafe. It is a very popular and unusual place.

The cafe offers visitors k different delicious desserts. The desserts are numbered from 0 to $k-1$. The cost of the i-th dessert is 2^i coins, because it is a binary cafe! Toma is willing to spend no more than n coins on tasting desserts. At the same time, he is not interested in buying any dessert more than once, because one is enough to evaluate the taste.

In how many different ways can he buy several desserts (possibly zero) for tasting?

Input Description

The first line of the input contains a single integer t ($1 \le t \le 1000$) — the number of test cases.

Then follows t lines, each of which describes one test case.

Each test case is given on a single line and consists of two integers n and k ($1 \le n, k \le 10^9$) — the number of coins Toma is willing to spend and the number of desserts in the binary cafe.

Output Description :

Output t integers, the i-th of which should be equal to the answer for the i-th test case — the number of ways to buy desserts for tasting.

Examples

Input
5 1 2 2 1 2 2 10 2 179 100
Output
2 2 3 4 180

C. Ski Resort

Constriants

Time Limit 1 seconds **Memory Limit** 256 MB

Problem Statement

Dima Vatrushin is a math teacher at school. He was sent on vacation for n days for his good work. Dima has long dreamed of going to a ski resort, so he wants to allocate several consecutive days and go skiing. Since the vacation requires careful preparation, he will only go for at least k days.

You are given an array a containing the weather forecast at the resort. That is, on the i-th day, the temperature will be a_i degrees.

Dima was born in Siberia, so he can go on vacation only if the temperature does not rise above q degrees throughout the vacation.

Unfortunately, Dima was so absorbed in abstract algebra that he forgot how to count. He asks you to help him and count the number of ways to choose vacation dates at the resort.

Input Description

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

Then follow the descriptions of the test cases.

The first line of each test case contains three integers n, k, q ($1 \le n \le 2 \cdot 10^5$, $1 \le k \le n$, $-10^9 \le q \le 10^9$) — the length of the array a, the minimum number of days at the resort, and the maximum comfortable temperature for Dima.

The second line of each test case contains n integers $a_1, a_2, a_3, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — the temperature at the ski resort.

The sum of all n values over all test cases does not exceed $2 \cdot 10^5$.

Output Description :

Output t integers, each of which is the answer to the corresponding test case — the number of ways for Dima to choose vacation dates at the resort.

Examples

Input
7 3 1 15 -5 0 -10 5 3 -33 8 12 9 0 5 4 3 12 12 12 10 15 4 1 -5 0 -1 2 5 5 5 0 3 -1 4 -5 -3 1 1 5 5 6 1 3 0 3 -2 5 -4 -4
Output
6 0 1 0 0 1 9

Note

In the first test case of the example, Dima can go on any day, so the suitable dates for him are [1], [2], [3], [1, 2], [2, 3], [1, 2, 3].
In the second and fourth test cases of the example, Dima cannot go on any day due to the high temperature, so there are no suitable dates.
In the third test case of the example, Dima can only go on the dates [1, 2, 3].

D. Wooden Toy Festival

Constriants

Time Limit 3 seconds **Memory Limit** 256 MB

Problem Statement

In a small town, there is a workshop specializing in woodwork. Since the town is small, only three carvers work there.

Soon, a wooden toy festival is planned in the town. The workshop employees want to prepare for it.

They know that n people will come to the workshop with a request to make a wooden toy. People are different and may want different toys. For simplicity, let's denote the pattern of the toy that the i-th person wants as a_i ($1 \le a_i \le 10^9$).

Each of the carvers can choose a pattern x ($1 \le x \le 10^9$) in advance, different carvers can choose different patterns. During the preparation for the festival, the carvers will perfectly work out the technique of making the toy of the chosen pattern, which will allow them to cut it out of wood instantly. To make a toy of pattern y for a carver who has chosen pattern x, it will take $|x - y|$ time, because the more the toy resembles the one he can make instantly, the faster the carver will cope with the work.

On the day of the festival, when the next person comes to the workshop with a request to make a wooden toy, the carvers can choose who will take on the job. At the same time, the carvers are very skilled people and can work on orders for different people simultaneously.

Since people don't like to wait, the carvers want to choose patterns for preparation in such a way that the maximum waiting time over all people is as small as possible.

Output the best maximum waiting time that the carvers can achieve.

Input Description

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

Then follow the descriptions of the test cases.

The first line of a test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of people who will come to the workshop.

The second line of a test case contains n integers $a_1, a_2, a_3, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the patterns of toys.

The sum of all n values over all test cases does not exceed $2 \cdot 10^5$.

Output Description :

Output t numbers, each of which is the answer to the corresponding test case — the best maximum waiting time that the carvers can achieve.

Examples

Input
5 6 1 7 7 9 9 9 6 5 4 2 1 30 60 9 14 19 37 59 1 4 4 98 73 1 2 6 3 10 1 17 15 11
Output
0 2 13 0 1

Note

In the first example, the carvers can choose patterns 1, 7, 9 for preparation.
In the second example, the carvers can choose patterns 3, 30, 60 for preparation.
In the thirs example, the carvers can choose patterns 14, 50, 85 for preparation.

E. Character Blocking

Constriants

Time Limit 2 seconds **Memory Limit** 256 MB

Problem Statement

You are given two strings of equal length s_1 and s_2, consisting of lowercase Latin letters, and an integer t.

You need to answer q queries, numbered from 1 to q. The i-th query comes in the i-th second of time. Each query is one of three types: block the characters at position pos (indexed from 1) in both strings for t seconds; swap two unblocked characters; determine if the two strings are equal at the time of the query, ignoring blocked characters.

Note that in queries of the second type, the characters being swapped can be from the same string or from s_1 and s_2.

Input Description

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

Then follow the descriptions of the test cases.

The first line of each test case contains a string s_1 consisting of lowercase Latin letters.

The second line of each test case contains a string s_2 consisting of lowercase Latin letters.

The strings have equal length.

The third line of each test case contains two integers t and q ($1 \le t, q \le 2 \cdot 10^5$). The number t indicates the number of seconds for which a character is blocked. The number q corresponds to the number of queries.

Each of the next q lines of each test case contains a single query. Each query is one of three types: "1 pos" — block the characters at position pos in both strings for t seconds; "2 1/2 pos_1 1/2 pos_2" — swap two unblocked characters. The second number in the query indicates the number of the string from which the first character for the swap is taken. The third number in the query indicates the position in that string of that character. The fourth number in the query indicates the number of the string from which the second character for the swap is taken. The fifth number in the query indicates the position in that string of that character; "3" — determine if the two strings are equal at the time of the query, ignoring blocked characters.

For queries of the first type, it is guaranteed that at the time of the query, the characters at position pos are not blocked.

For queries of the second type, it is guaranteed that the characters being swapped are not blocked.

All values of pos, pos_1, pos_2 are in the range from 1 to the length of the strings.

The sum of the values of q over all test cases, as well as the total length of the strings s_1, does not exceed $2 \cdot 10^5$.

Output Description :

For each query of the third type, output "YES" if the two strings s_1 and s_2 are equal at the time of the query, ignoring blocked characters, and "NO" otherwise.

You can 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.

Examples

Input
2 codeforces codeblocks 5 7 3 1 5 1 6 1 7 1 9 3 3 cool club 2 5 2 1 2 2 3 2 2 2 2 4 1 2 3 3
Output
NO YES NO YES NO

Note

Let's look at the strings s_1 and s_2 after each of the q queries. Blocked characters will be denoted in red.
First example input:
$(codeforces, codeblocks) \to (codeforces, codeblocks) \to (code\textcolor{red}{forces}, codeblocks) \to (code\textcolor{red}{forces}, \textcolor{red}{blocks}) \to (code\textcolor{red}{forces}, codeblocks) \to (code\textcolor{red}{forces}, codeblocks) \to (code\textcolor{red}{forces}, \textcolor{red}{blocks}) \to (code\textcolor{red}{forces}, code\textcolor{red}{blocks})$
Second example input:
$(cool, club) \to (cuol, clob) \to (cuol, cbol) \to (\textcolor{red}{cuol}, \textcolor{red}{cbol}) \to (\textcolor{red}{cuol}, \textcolor{red}{cbol}) \to (cuol, cbol)$

F. Railguns

Constriants

Time Limit 1 seconds **Memory Limit** 256 MB

Problem Statement

Tema is playing a very interesting computer game.

During the next mission, Tema's character found himself on an unfamiliar planet. Unlike Earth, this planet is flat and can be represented as an $n \times m$ rectangle.

Tema's character is located at the point with coordinates $(0, 0)$. In order to successfully complete the mission, he needs to reach the point with coordinates (n, m) alive.

Let the character of the computer game be located at the coordinate (i, j). Every second, starting from the first, Tema can: either use vertical hyperjump technology, after which his character will end up at coordinate $(i + 1, j)$ at the end of the second; or use horizontal hyperjump technology, after which his character will end up at coordinate $(i, j + 1)$ at the end of the second; or Tema can choose not to make a hyperjump, in which case his character will not move during this second;

The aliens that inhabit this planet are very dangerous and hostile. Therefore, they will shoot from their railguns r times.

Each shot completely penetrates one coordinate vertically or horizontally. If the character is in the line of its impact at the time of the shot (at the end of the second), he dies.

Since Tema looked at the game's source code, he knows complete information about each shot — the time, the penetrated coordinate, and the direction of the shot.

What is the minimum time for the character to reach the desired point? If he is doomed to die and cannot reach the point with coordinates (n, m), output -1.

Input Description

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

Then follow the descriptions of the test cases.

The first line of each test case contains two integers n and m ($1 \le n \cdot m \le 10^4$) — the size of the planet, its width and height.

The second line of each test case contains a single integer r ($1 \le r \le 100$) — the number of shots.

Then follow r lines, each describing one shot.

A shot is described by three integers t, d, *coord*. Where t is the second at which the shot will be fired ($1 \le t \le 10^9$). d is the direction of the shot ($d = 1$ denotes a horizontal shot, $d = 2$ denotes a vertical shot). *coord* is the size of the penetrated coordinate ($0 \le coord \le n$ for $d = 1$, $0 \le coord \le m$ for $d = 2$).

The sum of the products $n \cdot m$ over all test cases does not exceed 10^4.

Output Description :

For each test case, output a single number — the minimum time for the character to reach the coordinate (n, m), or -1 if he is doomed to die.

Examples

Input

```
5
1 3
4
1 2 0
2 2 1
3 2 2
4 1 1
3 3
6
2 1 0
2 1 1
2 1 2
2 2 0
2 2 1
2 2 2
2 1
3
7 1 2
2 1 1
7 2 1
2 2
5
9 1 2
3 2 0
5 1 2
4 2 2
7 1 0
4 6
7
6 1 2
12 1 3
4 1 0
17 2 3
1 2 6
16 2 6
3 2 4
```

Output

```
5
-1
3
6
10
```

Note

In the first test case, the character can move as follows: $(0,0) \to (0,1) \to (0,2) \to (0,3) \to (0,3) \to (1,3)$.

In the second test case, the character will not be able to leave the rectangle that will be completely penetrated by shots at the second 2.

G1. In Search of Truth (Easy Version)

Constriants

Time Limit 2 seconds **Memory Limit** 256 MB

Problem Statement

The only difference between easy and hard versions is the maximum number of queries. In this version, you are allowed to ask at most 2023 queries.

This is an interactive problem.

You are playing a game. The circle is divided into n sectors, sectors are numbered from 1 to n in some order. You are in the adjacent room and do not know either the number of sectors or their numbers. There is also an arrow that initially points to some sector. Initially, the host tells you the number of the sector to which the arrow points. After that, you can ask the host to move the arrow k sectors counterclockwise or clockwise at most 2023 times. And each time you are told the number of the sector to which the arrow points.

Your task is to determine the integer n — the number of sectors in at most 2023 queries.

It is guaranteed that $1 \leq n \leq 10^6$.

Input Description

The input consists of a single integer x ($1 \leq x \leq n$) — the number of the initial sector.

Output Description :

After you determine the integer n — the number of sectors, you should output "! n" ($1 \leq n \leq 10^6$). After that the program should immediately terminate.

Note that, printing the answer does not count as a query.

It is guaranteed that the integer n and the numbers of the sectors are fixed initially and will not be changed by the jury program depending on the queries.

Examples

Input
1 5 6 7 2 10 9 8 4 3 1
Output
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 ! 10

Note

Hacks
To hack, use the following test format.
In the first line, output a single integer n ($1 \leq n \leq 10^6$) — the number of sectors.
In the second line, output n different integers $1 \leq a_1, a_2, \ldots, a_n \leq n$ — the numbers of the sectors in clockwise order, the arrow initially points to the sector with the number a_1.

G2. In Search of Truth (Hard Version)

Constriants

Time Limit 2 seconds **Memory Limit** 256 MB

Problem Statement

The only difference between easy and hard versions is the maximum number of queries. In this version, you are allowed to ask at most 1000 queries.

This is an interactive problem.

You are playing a game. The circle is divided into n sectors, sectors are numbered from 1 to n in some order. You are in the adjacent room and do not know either the number of sectors or their numbers. There is also an arrow that initially points to some sector. Initially, the host tells you the number of the sector to which the arrow points. After that, you can ask the host to move the arrow k sectors counterclockwise or clockwise at most 1000 times. And each time you are told the number of the sector to which the arrow points.

Your task is to determine the integer n — the number of sectors in at most 1000 queries.

It is guaranteed that $1 \leq n \leq 10^6$.

Input Description

The input consists of a single integer x ($1 \leq x \leq n$) — the number of the initial sector.

Output Description :

After you determine the integer n — the number of sectors, you should output "! n" ($1 \leq n \leq 10^6$). After that the program should immediately terminate.

Note that, printing the answer does not count as a query.

It is guaranteed that the integer n and the numbers of the sectors are fixed initially and will not be changed by the jury program depending on the queries.

Examples

Input
1 5 6 7 2 10 9 8 4 3 1
Output
+ 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 ! 10

Note

Hacks
To hack, use the following test format.
In the first line, output a single integer n ($1 \leq n \leq 10^6$) — the number of sectors.
In the second line, output n different integers $1 \leq a_1, a_2, \ldots, a_n \leq n$ — the numbers of the sectors in clockwise order, the arrow initially points to the sector with the number a_1.