Problem A. Registration System

Time limit	5000 ms
Mem limit	65536 kB
Input file	stdin
Output file	stdout

A new e-mail service "Berlandesk" is going to be opened in Berland in the near future. The site administration wants to launch their project as soon as possible, that's why they ask you to help. You're suggested to implement the prototype of site registration system. The system should work on the following principle.

Each time a new user wants to register, he sends to the system a request with his `name`. If such a `name` does not exist in the system database, it is inserted into the database, and the user gets the response `OK`, confirming the successful registration. If the `name` already exists in the system database, the system makes up a new user name, sends it to the user as a prompt and *also inserts the prompt into the database.* The new name is formed by the following rule. Numbers, starting with 1, are appended one after another to `name` (`name1`, `name2`, ...), among these numbers the least i is found so that `name`i does not yet exist in the database.

Input

The first line contains number n ($1 \le n \le 10^5$). The following n lines contain the requests to the system. Each request is a non-empty line, and consists of not more than 32 characters, which are all lowercase Latin letters.

Output

Print n lines, which are system responses to the requests: `OK` in case of successful registration, or a prompt with a new name, if the requested name is already taken.

Examples

Input	Output
4 abacaba acaba abacaba acab	OK OK abacaba1 OK

Input	Output
6 first first second second third third	OK first1 OK second1 OK third1

Jack and Jill have decided to sell some of their Compact Discs, while they still have some value. They have decided to sell one of each of the CD titles that they both own. How many CDs can Jack and Jill sell?

Neither Jack nor Jill owns more than one copy of each CD.

Input

The input consists of a sequence of test cases. The first line of each test case contains two non-negative integers N and M, each at most one million, specifying the number of CDs owned by Jack and by Jill, respectively. This line is followed by N lines listing the catalog numbers of the CDs owned by Jack in increasing order, and M more lines listing the catalog numbers of the CDs owned by Jill in increasing order. Each catalog number is a positive integer no greater than one billion. The input is terminated by a line containing two zeros. This last line is not a test case and should not be processed.

Output

For each test case, output a line containing one integer, the number of CDs that Jack and Jill both own.

Sample Input

```
3 3
1
2
3
1
2
4
0 0
```

Sample Output

```
2
```

Problem C. Distinct Numbers

Time limit 1000 ms
Mem limit 524288 kB

You are given a list of n integers, and your task is to calculate the number of *distinct* values in the list.

Input

The first input line has an integer n: the number of values.

The second line has n integers $x_1, x_2, \ldots, x_n$.

Output

Print one integers: the number of distinct values.

Constraints

- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq x_i \leq 10^9$

Example

Input	Output
5 2 3 2 2 3	2

Problem D. Sereja and Suffixes

Time limit	1000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

Sereja has an array a, consisting of n integers $a_1, a_2, ..., a_n$. The boy cannot sit and do nothing, he decided to study an array. Sereja took a piece of paper and wrote out m integers $l_1, l_2, ..., l_m$ $(1 \le l_i \le n)$. For each number l_i he wants to know how many distinct numbers are staying on the positions $l_i, l_i + 1, ..., n$. Formally, he want to find the number of distinct numbers among $a_{l_i}, a_{l_i+1}, ..., a_n$.?

Sereja wrote out the necessary array elements but the array was so large and the boy was so pressed for time. Help him, find the answer for the described question for each l_i.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^5)$. The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^5)$ — the array elements.

Next m lines contain integers $l_1, l_2, ..., l_m$. The i-th line contains integer l_i $(1 \le l_i \le n)$.

Output

Print m lines — on the i-th line print the answer to the number l_i.

Examples

Input	Output
10 10 1 2 3 4 1 2 3 4 100000 99999 1 2 3 4 5 6 7 8 9 10	6 6 6 6 6 5 4 3 2 1

Problem E. Odd Queries

Time limit 2000 ms
Mem limit 262144 kB

You have an array $a_1, a_2, \ldots, a_n$. Answer q queries of the following form:

- If we change all elements in the range $a_l, a_{l+1}, \ldots, a_r$ of the array to k, will the sum of the entire array be odd?

Note that queries are **independent** and do not affect future queries.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case consists of 2 integers n and q ($1 \le n \le 2 \cdot 10^5$; $1 \le q \le 2 \cdot 10^5$) — the length of the array and the number of queries.

The second line of each test case consists of n integers a_i ($1 \le a_i \le 10^9$) — the array a.

The next q lines of each test case consists of 3 integers l, r, k ($1 \le l \le r \le n$; $1 \le k \le 10^9$) — the queries.

It is guaranteed that the sum of n over all test cases doesn't exceed $2 \cdot 10^5$, and the sum of q doesn't exceed $2 \cdot 10^5$.

Output

For each query, output "`YES`" if the sum of the entire array becomes odd, and "`NO`" otherwise.

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

Examples

Input	Output
2 5 5 2 2 1 3 2 2 3 3 2 3 4 1 5 5 1 4 9 2 4 3 10 5 1 1 1 1 1 1 1 1 1 1 3 8 13 2 5 10 3 8 10 1 10 2 1 9 100	YES YES YES NO YES NO NO NO NO YES

Note

For the first test case:

- If the elements in the range $(2, 3)$ would get set to 3 the array would become $\{2, 3, 3, 3, 2\}$, the sum would be $2 + 3 + 3 + 3 + 2 = 13$ which is odd, so the answer is "YES".
- If the elements in the range $(2, 3)$ would get set to 4 the array would become $\{2, 4, 4, 3, 2\}$, the sum would be $2 + 4 + 4 + 3 + 2 = 15$ which is odd, so the answer is "YES".
- If the elements in the range $(1, 5)$ would get set to 5 the array would become $\{5, 5, 5, 5, 5\}$, the sum would be $5 + 5 + 5 + 5 + 5 = 25$ which is odd, so the answer is "YES".
- If the elements in the range $(1, 4)$ would get set to 9 the array would become $\{9, 9, 9, 9, 2\}$, the sum would be $9 + 9 + 9 + 9 + 2 = 38$ which is even, so the answer is "NO".
- If the elements in the range $(2, 4)$ would get set to 3 the array would become $\{2, 3, 3, 3, 2\}$, the sum would be $2 + 3 + 3 + 3 + 2 = 13$ which is odd, so the answer is "YES".

Problem F. Word Game

Time limit 1000 ms
Mem limit 262144 kB

Three guys play a game: first, each person writes down n distinct words of length 3. Then, they total up the number of points as follows:

- if a word was written by one person — that person gets 3 points,
- if a word was written by two people — each of the two gets 1 point,
- if a word was written by all — nobody gets any points.

In the end, how many points does each player have?

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 100$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n ($1 \leq n \leq 1000$) — the number of words written by each person.

The following three lines each contain n **distinct** strings — the words written by each person. Each string consists of 3 lowercase English characters.

Output

For each test case, output three space-separated integers — the number of points each of the three guys earned. You should output the answers in the same order as the input; the i-th integer should be the number of points earned by the i-th guy.

Examples

Input	Output
3 1 abc def abc 3 orz for qaq qaq orz for cod for ces 5 iat roc hem ica lly bac ter iol ogi sts bac roc lly iol iat	1 3 1 2 2 6 9 11 5

Note

In the first test case:

- The word `abc` was written by the first and third guys — they each get 1 point.
- The word `def` was written by the second guy only — he gets 3 points.

Problem G. Queries about less or equal elements

Time limit 2000 ms
Mem limit 262144 kB

You are given two arrays of integers a and b. For each element of the second array b_j you should find the number of elements in array a that are less than or equal to the value b_j.

Input

The first line contains two integers n, m ($1 \le n, m \le 2 \cdot 10^5$) — the sizes of arrays a and b.

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

The third line contains m integers — the elements of array b ($-10^9 \le b_j \le 10^9$).

Output

Print m integers, separated by spaces: the j-th of which is equal to the number of such elements in array a that are less than or equal to the value b_j.

Examples

Input	Output
5 4 1 3 5 7 9 6 4 2 8	3 2 1 4

Input	Output
5 5 1 2 1 2 5 3 1 4 1 5	4 2 4 2 5

Problem H. Plug-in

Time limit	1000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

Polycarp thinks about the meaning of life very often. He does this constantly, even when typing in the editor. Every time he starts brooding he can no longer fully concentrate and repeatedly presses the keys that need to be pressed only once. For example, instead of the phrase "how are you" he can type "hhoow aaaare yyoouu".

Polycarp decided to automate the process of correcting such errors. He decided to write a plug-in to the text editor that will remove pairs of identical consecutive letters (if there are any in the text). Of course, this is not exactly what Polycarp needs, but he's got to start from something!

Help Polycarp and write the main plug-in module. Your program should remove from a string all pairs of identical letters, which are consecutive. If after the removal there appear new pairs, the program should remove them as well. Technically, its work should be equivalent to the following: while the string contains a pair of consecutive identical letters, the pair should be deleted. Note that deleting of the consecutive identical letters can be done in any order, as any order leads to the same result.

Input

The input data consists of a single line to be processed. The length of the line is from 1 to $2 \cdot 10^5$ characters inclusive. The string contains only lowercase Latin letters.

Output

Print the given string after it is processed. It is guaranteed that the result will contain at least one character.

Examples

Input	Output
hhoowaaaareyyoouu	wre

Input	Output
reallazy	rezy

Input	Output
abacabaabacabaa	a

Problem I. Musical Puzzle

Time limit 1000 ms
Mem limit 262144 kB

Vlad decided to compose a melody on his guitar. Let's represent the melody as a sequence of notes corresponding to the characters 'a', 'b', 'c', 'd', 'e', 'f', and 'g'.

However, Vlad is not very experienced in playing the guitar and can only record **exactly two** notes at a time. Vlad wants to obtain the melody s, and to do this, he can merge the recorded melodies together. In this case, the last sound of the first melody must match the first sound of the second melody.

For example, if Vlad recorded the melodies `"ab"` and `"ba"`, he can merge them together and obtain the melody `"aba"`, and then merge the result with `"ab"` to get `"abab"`.

Help Vlad determine the **minimum** number of melodies consisting of two notes that he needs to record in order to obtain the melody s.

Input

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

Following that are the descriptions of the test cases.

The first line of each test case contains an integer n ($2 \leq n \leq 50$) — the length of the melody s.

The second line of each test case contains a string s of length n, consisting of characters 'a', 'b', 'c', 'd', 'e', 'f', 'g'.

Output

Output t integers, each representing the answer for the corresponding test case. As the answer output **minimum** number of melodies consisting of two notes that Vlad needs to record.

Examples

Input	Output
5 4 abab 7 abacaba 6 aaaaaa 7 abcdefg 5 babdd	2 4 1 6 4

Note

In the first sample, you need to record the melodies "ab" and "ba", as described in the problem statement.

In the second sample, you need to record the melodies "ab", "ba", "ac", and "ca".

In the third sample, the only necessary melody is "aa".

Problem J. Minimum Stocks

Time limit	500 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

Himanshu wants to invest into stock market and his friend Navneet helps him by providing him instruction for next *N* days.

Navneet gives Himanshu 3 types of instruction,

1 *X Y* There is a stock X available at price Y. Here X is a string and Y is an integer.

2 *X Z* The price of stock X has changed to Z. Here X is a string and Z is an integer.

3 BUY Buy the stock which has the lowest price.

You as a programmer, are given all the instructions of ***N*** days. Can you tell, **which stock did Himanshu buy on which day**. Print the output in same order as Himanshu bought the stock. See sample input and output for clarification.

At any point of time, there is **atmost one stock of *X***. However, *X* can be made available to market again through another instruction of type 1.

All instructions are valid. i.e. There is always some stock to buy having the minimum price of all. Also if the price of *X* has changed, then *X* is already known and hasn't been bought yet.

Input

First line contains *N*. ($1 \le N \le 10^6$)

Next *N* lines, each of them contains an instruction of any of 3 types. (Look at instruction format above.)

In any instruction, (X is a string of length upto 10 characters. All characters are from English alphabet, both small and capital), and ($0 \le Y \le 10^9$) and ($0 \le Z \le 10^9$).

Output

For each instruction of type 3, output two values *X* and *Y*. Where *X* is the name of Stock having minimum price and *Y* is the day on which it was bought.

Example

Input	Output
7 1 ABC 32 1 XDC 54 3 BUY 1 XCD 32 1 ABC 12 2 XDC 10 3 BUY	ABC 3 XDC 7

Explanation

On day 3, there is instruction to buy. There are two stocks available "XDC" and "ABC", since price of "ABC" is less, he buys it. After this "ABC" is not available in market anymore.

On day 7, there is instruction to buy. Of all stocks available, "XDC" has the least price and hence he buys "XDC".

Problem K. Cellular Network

Time limit 3000 ms
Mem limit 262144 kB

You are given n points on the straight line — the positions (x-coordinates) of the cities and m points on the same line — the positions (x-coordinates) of the cellular towers. All towers work in the same way — they provide cellular network for all cities, which are located at the distance which is no more than r from this tower.

Your task is to find minimal r that each city has been provided by cellular network, i.e. for each city there is at least one cellular tower at the distance which is no more than r.

If $r = 0$ then a tower provides cellular network only for the point where it is located. One tower can provide cellular network for any number of cities, but all these cities must be at the distance which is no more than r from this tower.

Input

The first line contains two positive integers n and m ($1 \le n, m \le 10^5$) — the number of cities and the number of cellular towers.

The second line contains a sequence of n integers $a_1, a_2, ..., a_n$ ($-10^9 \le a_i \le 10^9$) — the coordinates of cities. It is allowed that there are any number of cities in the same point. All coordinates a_i are given in non-decreasing order.

The third line contains a sequence of m integers $b_1, b_2, ..., b_m$ ($-10^9 \le b_j \le 10^9$) — the coordinates of cellular towers. It is allowed that there are any number of towers in the same point. All coordinates b_j are given in non-decreasing order.

Output

Print minimal r so that each city will be covered by cellular network.

Examples

Input	Output
3 2 -2 2 4 -3 0	4

Input	Output
5 3 1 5 10 14 17 4 11 15	3

Problem L. Cutting Woods

Time limit 2000 ms
Mem limit 1048576 kB

Problem Statement

We have a long piece of timber with a length of L meters.
For each $x = 1, 2, \ldots, L-1$, there is a mark called Mark x at x meters from the left end of the piece.

You are given Q queries, the i-th of which is represented as a pair of numbers (c_i, x_i).
Process the queries in ascending order of i as described below.

- If $c_i = 1$: cut the piece at Mark x_i into two.
- If $c_i = 2$: choose the piece with Mark x_i on it and print its length.

Here, for both kinds of queries $c_i = 1, 2$, it is guaranteed that there will have been no cut at Mark x_i when the query is to be processed.

Constraints

- $1 \leq L \leq 10^9$
- $1 \leq Q \leq 2 \times 10^5$
- $c_i = 1, 2 \ (1 \leq i \leq Q)$
- $1 \leq x_i \leq L-1 \ (1 \leq i \leq Q)$
- For every $i \ (1 \leq i \leq Q)$, the following holds: there is no j such that $1 \leq j < i$ and $(c_j, x_j) = (1, x_i)$.
- All values in input are integers.

Input

Input is given from Standard Input in the following format:

L Q
c_1 x_1
c_2 x_2

$\vdots$
$c_Q \ x_Q$

Output

Print the number of lines equal to the number of queries $c_i = 2$. In the j-th line, print the response to the j-th such query.

Sample 1

Input	Output
5 3 2 2 1 3 2 2	5 3

At the time of the first query, no cut has been made, so the piece with Mark 2 has a length of 5 meters. Thus, you should print 5.
In the second query, the piece is cut into two pieces with lengths of 3 and 2 meters.
At the time of the third query, the piece with Mark 2 has a length of 3 meters, so you should print 3.

Sample 2

Input	Output
5 3 1 2 1 4 2 3	2

Sample 3

Input	Output
100 10 1 31 2 41 1 59 2 26 1 53 2 58 1 97 2 93 1 23 2 84	69 31 6 38 38

Problem M. YetnotherrokenKeoard

Time limit 1000 ms
Mem limit 262144 kB

Polycarp has a problem — his laptop keyboard is broken.

Now, when he presses the '`b`' key, it acts like an unusual backspace: it deletes the last (rightmost) lowercase letter in the typed string. If there are no lowercase letters in the typed string, then the press is completely ignored.

Similarly, when he presses the '`B`' key, it deletes the last (rightmost) uppercase letter in the typed string. If there are no uppercase letters in the typed string, then the press is completely ignored.

In both cases, the letters '`b`' and '`B`' are not added to the typed string when these keys are pressed.

Consider an example where the sequence of key presses was "`ARaBbbitBaby`". In this case, the typed string will change as follows: "" $\xrightarrow{\texttt{A}}$ "`A`" $\xrightarrow{\texttt{R}}$ "`AR`" $\xrightarrow{\texttt{a}}$ "`ARa`" $\xrightarrow{\texttt{B}}$ "`Aa`" $\xrightarrow{\texttt{b}}$ "`A`" $\xrightarrow{\texttt{b}}$ "`A`" $\xrightarrow{\texttt{i}}$ "`Ai`" $\xrightarrow{\texttt{t}}$ "`Ait`" $\xrightarrow{\texttt{B}}$ "`it`" $\xrightarrow{\texttt{a}}$ "`ita`" $\xrightarrow{\texttt{b}}$ "`it`" $\xrightarrow{\texttt{y}}$ "`ity`".

Given a sequence of pressed keys, output the typed string after processing all key presses.

Input

The first line of the input data contains an integer t ($1 \le t \le 1000$), the number of test cases in the test.

The following contains t non-empty lines, which consist of lowercase and uppercase letters of the Latin alphabet.

It is guaranteed that each line contains at least one letter and the sum of the lengths of the lines does not exceed 10^6.

Output

For each test case, output the result of processing the key presses on a separate line. If the typed string is empty, then output an empty line.

Examples

Input	Output
12 ARaBbbitBaby YetAnotherBrokenKeyboard Bubble Improbable abbreviable BbBB BusyasaBeeinaBedofBloomingBlossoms CoDEBARbIES codeforces bobebobbes b TheBBlackbboard	ity YetnotherrokenKeoard le Imprle revile usyasaeeinaedofloominglossoms CDARIES codeforces es helaoard

Problem N. Chimpanzini Bananini

Time limit 2000 ms
Mem limit 262144 kB

Chimpanzini Bananini stands on the brink of a momentous battle—one destined to bring finality.

For an arbitrary array b of length m, let's denote the *rizziness* of the array to be $\sum_{i=1}^{m} b_i \cdot i = b_1 \cdot 1 + b_2 \cdot 2 + b_3 \cdot 3 + \ldots + b_m \cdot m$.

Chimpanzini Bananini gifts you an empty array. There are three types of operations you can perform on it.

1. Perform a cyclic shift on the array. That is, the array $[a_1, a_2, \ldots, a_n]$ becomes $[a_n, a_1, a_2, \ldots, a_{n-1}]$.
2. Reverse the entire array. That is, the array $[a_1, a_2, \ldots, a_n]$ becomes $[a_n, a_{n-1}, \ldots, a_1]$.
3. Append an element to the end of the array. The array $[a_1, a_2, \ldots, a_n]$ becomes $[a_1, a_2, \ldots, a_n, k]$ after appending k to the end of the array.

After each operation, you are interested in calculating the *rizziness* of your array.

Note that all operations are **persistent**. This means that each operation modifies the array, and subsequent operations should be applied to the current state of the array after the previous operations.

Input

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

The first line of the input contains an integer q ($1 \le q \le 2 \cdot 10^5$) — the number of operations you perform on your array.

The following q lines first contain a single integer s ($1 \le s \le 3$) — the operation type.

- If $s = 1$, then the cyclic shift operation should be performed.
- If $s = 2$, then the reversal operation should be performed.
- If $s = 3$, then the line will contain an additional integer k ($1 \le k \le 10^6$), denoting the element appended to the back of the array.

It is guaranteed that the sum of q will not exceed $2 \cdot 10^5$ over all test cases. Additionally, it is guaranteed that the first operation on each test case will be one with $s = 3$.

Output

For each test case, output q lines, outputting the *rizziness* of your array after each operation.

Examples

Input	Output
1 13 3 1 3 2 3 3 1 3 4 2 3 5 1 3 6 2 3 7 2 1	1 5 14 11 27 23 48 38 74 73 122 102 88

Note

The first six states of the array:

- $[1]$
- $[1, 2]$
- $[1, 2, 3]$
- $[3, 1, 2]$
- $[3, 1, 2, 4]$
- $[4, 2, 1, 3]$