Educational Codeforces Round 130 (Rated for Div. 2)

A. Parkway Walk

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

You are walking through a parkway near your house. The parkway has $n + 1$ benches in a row numbered from 1 to $n + 1$ from left to right. The distance between the bench i and $i + 1$ is a_i meters.

Initially, you have m units of energy. To walk 1 meter of distance, you spend 1 unit of your energy. You can't walk if you have no energy. Also, you can restore your energy by **sitting on benches** (and this is the only way to restore the energy). When you are sitting, you can restore any integer amount of energy you want (if you sit longer, you restore more energy). Note that the amount of your energy **can exceed** m.

Your task is to find the **minimum** amount of energy you have to **restore** (by sitting on benches) to reach the bench $n + 1$ from the bench 1 (and end your walk).

You have to answer t independent test cases.

Input

The first line of the input contains one integer t ($1 \le t \le 100$) — the number of test cases. Then t test cases follow.

The first line of the test case contains two integers n and m ($1 \le n \le 100$; $1 \le m \le 10^4$).

The second line of the test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 100$), where a_i is the distance between benches i and $i + 1$.

Output

For each test case, print one integer — the **minimum** amount of energy you have to **restore** (by sitting on benches) to reach the bench $n + 1$ from the bench 1 (and end your walk) in the corresponding test case.

Example

input
3 3 1 1 2 1 4 5 3 3 5 2 5 16 1 2 3 4 5
output
3 8 0

Note

In the first test case of the example, you can walk to the bench 2, spending 1 unit of energy, then restore 2 units of energy on the second bench, walk to the bench 3, spending 2 units of energy, restore 1 unit of energy and go to the bench 4.

In the third test case of the example, you have enough energy to just go to the bench 6 without sitting at all.

B. Promo

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

The store sells n items, the price of the i-th item is p_i. The store's management is going to hold a promotion: if a customer purchases at least x items, y cheapest of them are free.

The management has not yet decided on the exact values of x and y. Therefore, they ask you to process q queries: for the given values of x and y, determine the maximum total value of items received for free, if a customer makes **one purchase**.

Note that all queries are independent; they don't affect the store's stock.

Input

The first line contains two integers n and q ($1 \le n, q \le 2 \cdot 10^5$) — the number of items in the store and the number of queries,

respectively.

The second line contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 10^6$), where p_i — the price of the i-th item.

The following q lines contain two integers x_i and y_i each ($1 \le y_i \le x_i \le n$) — the values of the parameters x and y in the i-th query.

Output
For each query, print a single integer — the maximum total value of items received for free **for one purchase**.

Example

input
5 3 5 3 1 5 2 3 2 1 1 5 3
output
8 5 6

Note
In the first query, a customer can buy three items worth $5, 3, 5$, the two cheapest of them are $3 + 5 = 8$.

In the second query, a customer can buy two items worth 5 and 5, the cheapest of them is 5.

In the third query, a customer has to buy all the items to receive the three cheapest of them for free; their total price is $1 + 2 + 3 = 6$.

C. awoo's Favorite Problem

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

You are given two strings s and t, both of length n. Each character in both string is 'a', 'b' or 'c'.

In one move, you can perform one of the following actions:

- choose an occurrence of "ab" in s and replace it with "ba";
- choose an occurrence of "bc" in s and replace it with "cb".

You are allowed to perform an arbitrary amount of moves (possibly, zero). Can you change string s to make it equal to string t?

Input
The first line contains a single integer q ($1 \le q \le 10^4$) — the number of testcases.

The first line of each testcase contains a single integer n ($1 \le n \le 10^5$) — the length of strings s and t.

The second line contains string s of length n. Each character is 'a', 'b' or 'c'.

The third line contains string t of length n. Each character is 'a', 'b' or 'c'.

The sum of n over all testcases doesn't exceed 10^5.

Output
For each testcase, print "YES" if you can change string s to make it equal to string t by performing an arbitrary amount of moves (possibly, zero). Otherwise, print "NO".

Example

input
5 3 cab cab 1 a b 6 abbabc bbaacb 10 bcaababababc cbbababaac 2 ba ab

output
YES NO YES YES NO

D. Guess The String

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

This is an interactive problem. Remember to flush your output while communicating with the testing program. You may use `fflush(stdout)` in C++, `system.out.flush()` in Java, `stdout.flush()` in Python or `flush(output)` in Pascal to flush the output. If you use some other programming language, consult its documentation. You may also refer to the guide on interactive problems: https://codeforces.com/blog/entry/45307.

The jury has chosen a string s consisting of n characters; each character of s is a lowercase Latin letter. Your task is to guess this string; initially, you know only its length.

You may ask queries of two types:

- $1\ i$ — the query of the first type, where i is an integer from 1 to n. In response to this query, the jury will tell you the character s_i;
- $2\ l\ r$ — the query of the second type, where l and r are integers such that $1 \le l \le r \le n$. In response to this query, the jury will tell you the number of different characters among $s_l, s_{l+1}, \ldots, s_r$.

You are allowed to ask no more than 26 queries of the first type, and no more than 6000 queries of the second type. Your task is to restore the string s.

For each test in this problem, the string s is fixed beforehand, and will be the same for every submission.

Input
Initially, the jury program sends one integer n on a separate line — the size of s ($1 \le n \le 1000$).

Output
To give the answer, print one line `! s` *with a line break in the end*, where s should be the string picked by the jury. After that, your program should flush the output and terminate gracefully.

Interaction
To ask a query, you should send one line containing the query, in one of the following formats:

- `? 1 i` — for a query of the first type ($1 \le i \le n$);
- `? 2 l r` — for a query of the second type ($1 \le l \le r \le n$).

Don't forget to flush the output after sending the query line.

The answer to your query will be given on a separate line. For a query of the first type, the answer will be the character s_i. For a query of the second type, the answer will be an integer equal to the number of different characters among $s_l, s_{l+1}, \ldots, s_r$.

You are allowed to ask no more than 26 queries of the first type, and no more than 6000 queries of the second type.

In case you ask too many queries, or the jury program fails to recognize your query format, the answer to your query will be one integer 0. After receiving 0 as the answer, your program should terminate immediately — otherwise you may receive verdict "Runtime error", "Time limit exceeded" or some other verdict instead of "Wrong answer".

Example

input
5 4 u 2 g e s 1

output
? 2 1 5 ? 1 2 ? 2 1 2 ? 1 1 ? 1 3 ? 1 4 ? 2 4 5 ! guess

Note
Let's analyze the example of interaction.

The string chosen by the jury is guess, so initially the jury sends one integer 5.

1. the first query is ? 2 1 5, which means "count the number of different characters among $s_1, s_2, \ldots, s_5$". The answer to it is 4.
2. the second query is ? 1 2, which means "tell which character is s_2". The answer to it is u.
3. the third query is ? 2 1 2, which means "count the number of different characters among s_1 and s_2". The answer to it is 2.
4. the fourth query is ? 1 1, which means "tell which character is s_1". The answer to it is g.
5. the fifth query is ? 1 3, which means "tell which character is s_3". The answer to it is e.
6. the sixth query is ? 1 4, which means "tell which character is s_4". The answer to it is s.
7. the seventh query is ? 2 4 5, which means "count the number of different characters among s_4 and s_5". The answer to it is 1, so it's possible to deduce that s_4 is the same as s_5.

In the end, the answer is submitted as ! guess, and it is deduced correctly.

E. Coloring

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given n points on the plane, the coordinates of the i-th point are (x_i, y_i). No two points have the same coordinates.

The distance between points i and j is defined as $d(i, j) = |x_i - x_j| + |y_i - y_j|$.

For each point, you have to choose a color, represented by an integer from 1 to n. For every ordered triple of different points (a, b, c), the following constraints should be met:

- if a, b and c have the same color, then $d(a, b) = d(a, c) = d(b, c)$;
- if a and b have the same color, and the color of c is different from the color of a, then $d(a, b) < d(a, c)$ and $d(a, b) < d(b, c)$.

Calculate the number of different ways to choose the colors that meet these constraints.

Input
The first line contains one integer n ($2 \le n \le 100$) — the number of points.

Then n lines follow. The i-th of them contains two integers x_i and y_i ($0 \le x_i, y_i \le 10^8$).

No two points have the same coordinates (i. e. if $i \ne j$, then either $x_i \ne x_j$ or $y_i \ne y_j$).

Output
Print one integer — the number of ways to choose the colors for the points. Since it can be large, print it modulo 998244353.

Examples

input
3 1 0 3 0 2 1
output
9

input
5 1 2 2 4 3 4 4 4 1 3
output
240

input
4 1 0 3 0 2 1 2 0
output
24

Note

In the first test, the following ways to choose the colors are suitable:

- $[1, 1, 1]$;
- $[2, 2, 2]$;
- $[3, 3, 3]$;
- $[1, 2, 3]$;
- $[1, 3, 2]$;
- $[2, 1, 3]$;
- $[2, 3, 1]$;
- $[3, 1, 2]$;
- $[3, 2, 1]$.

F. Too Many Constraints

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are asked to build an array a, consisting of n integers, each element should be from 1 to k.

The array should be non-decreasing ($a_i \le a_{i+1}$ for all i from 1 to $n-1$).

You are also given additional constraints on it. Each constraint is of one of three following types:

- $1\ i\ x$: a_i **should not** be equal to x;
- $2\ i\ j\ x$: $a_i + a_j$ should be less than or equal to x;
- $3\ i\ j\ x$: $a_i + a_j$ should be greater than or equal to x.

Build any non-decreasing array that satisfies all constraints or report that no such array exists.

Input

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

The first line of each testcase contains three integers n, m and k ($2 \le n \le 2 \cdot 10^4$; $0 \le m \le 2 \cdot 10^4$; $2 \le k \le 10$).

The i-th of the next m lines contains a description of a constraint. Each constraint is of one of three following types:

- $1\ i\ x$ ($1 \le i \le n$; $1 \le x \le k$): a_i **should not** be equal to x;
- $2\ i\ j\ x$ ($1 \le i < j \le n$; $2 \le x \le 2 \cdot k$): $a_i + a_j$ should be less than or equal to x;
- $3\ i\ j\ x$ ($1 \le i < j \le n$; $2 \le x \le 2 \cdot k$): $a_i + a_j$ should be greater than or equal to x.

The sum of n over all testcases doesn't exceed $2 \cdot 10^4$. The sum of m over all testcases doesn't exceed $2 \cdot 10^4$.

Output

For each testcase, determine if there exists a non-decreasing array that satisfies all conditions. If there is no such array, then print -1. Otherwise, print any valid array — n integers from 1 to k.

Example

input
4 4 0 4 2 2 3 3 1 2 3 1 2 2 3 3 2 1 1 1 2 2 3 2 3 2 3 2 5 5 5 3 2 5 7 2 4 5 10 3 4 5 6 3 3 4 7 2 1 5 7
output
1 2 3 4 1 3 -1 1 2 2 5 5