

Codeforces Round #368 (Div. 2)

A. Brain's Photos

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Small, but very brave, mouse Brain was not accepted to summer school of young villains. He was upset and decided to postpone his plans of taking over the world, but to become a photographer instead.

As you may know, the coolest photos are on the film (because you can specify the hashtag #film for such).

Brain took a lot of colourful pictures on colored and black-and-white film. Then he developed and translated it into a digital form. But now, color and black-and-white photos are in one folder, and to sort them, one needs to spend more than one hour!

As soon as Brain is a photographer not programmer now, he asks you to help him determine for a **single** photo whether it is colored or black-and-white.

Photo can be represented as a matrix sized $n \times m$, and each element of the matrix stores a symbol indicating corresponding pixel color. There are only 6 colors:

- 'C' (cyan)
- 'M' (magenta)
- 'Y' (yellow)
- 'W' (white)
- 'G' (grey)
- 'B' (black)

The photo is considered black-and-white if it has only white, black and grey pixels in it. If there are any of cyan, magenta or yellow pixels in the photo then it is considered colored.

Input

The first line of the input contains two integers n and m ($1 \leq n, m \leq 100$) — the number of photo pixel matrix rows and columns respectively.

Then n lines describing matrix rows follow. Each of them contains m space-separated characters describing colors of pixels in a row. Each character in the line is one of the 'C', 'M', 'Y', 'W', 'G' or 'B'.

Output

Print the "#Black&White" (without quotes), if the photo is black-and-white and "#Color" (without quotes), if it is colored, in the only line.

Examples

input
2 2 C M Y Y
output
#Color
input
3 2 W W W W B B
output
#Black&White
input
1 1 W
output
#Black&White

B. Bakery

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Masha wants to open her own bakery and bake muffins in one of the n cities numbered from 1 to n . There are m bidirectional roads, each of whose connects some pair of cities.

To bake muffins in her bakery, Masha needs to establish flour supply from some storage. There are only k storages, located in different cities numbered a_1, a_2, \dots, a_k .

Unfortunately the law of the country Masha lives in prohibits opening bakery in any of the cities which has storage located in it. She can open it only in one of another $n - k$ cities, and, of course, flour delivery should be paid — for every kilometer of path between storage and bakery Masha should pay 1 ruble.

Formally, Masha will pay x roubles, if she will open the bakery in some city b ($a_i \neq b$ for every $1 \leq i \leq k$) and choose a storage in some city s ($s = a_j$ for some $1 \leq j \leq k$) and b and s are connected by some path of roads of summary length x (if there are more than one path, Masha is able to choose which of them should be used).

Masha is very thrifty and rational. She is interested in a city, where she can open her bakery (and choose one of k storages and one of the paths between city with bakery and city with storage) and pay minimum possible amount of rubles for flour delivery. Please help Masha find this amount.

Input

The first line of the input contains three integers n, m and k ($1 \leq n, m \leq 10^5, 0 \leq k \leq n$) — the number of cities in country Masha lives in, the number of roads between them and the number of flour storages respectively.

Then m lines follow. Each of them contains three integers u, v and l ($1 \leq u, v \leq n, 1 \leq l \leq 10^9, u \neq v$) meaning that there is a road between cities u and v of length of l kilometers.

If $k > 0$, then the last line of the input contains k distinct integers a_1, a_2, \dots, a_k ($1 \leq a_i \leq n$) — the number of cities having flour storage located in. If $k = 0$ then this line is **not presented in the input**.

Output

Print the minimum possible amount of rubles Masha should pay for flour delivery in the only line.

If the bakery can not be opened (while satisfying conditions) in any of the n cities, print - 1 in the only line.

Examples

input
5 4 2 1 2 5 1 2 3 2 3 4 1 4 10 1 5
output
3
input
3 1 1 1 2 3 3
output
-1

Note

Image illustrates the first sample case. Cities with storage located in and the road representing the answer are darkened.

C. Pythagorean Triples

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

Katya studies in a fifth grade. Recently her class studied right triangles and the Pythagorean theorem. It appeared, that there are triples of positive integers such that you can construct a right triangle with segments of lengths corresponding to triple. Such triples are called *Pythagorean triples*.

For example, triples (3, 4, 5), (5, 12, 13) and (6, 8, 10) are Pythagorean triples.

Here Katya wondered if she can specify the length of some side of right triangle and find any Pythagorean triple corresponding to such length? Note that the side which length is specified can be a cathetus as well as hypotenuse.

Katya had no problems with completing this task. Will you do the same?

Input

The only line of the input contains single integer n ($1 \leq n \leq 10^9$) — the length of some side of a right triangle.

Output

Print two integers m and k ($1 \leq m, k \leq 10^{18}$), such that n , m and k form a Pythagorean triple, in the only line.

In case if there is no any Pythagorean triple containing integer n , print - 1 in the only line. If there are many answers, print any of them.

Examples

input
3
output
4 5

input
6
output
8 10

input
1
output
-1

input
17
output
144 145

input
67
output
2244 2245

Note

Illustration for the first sample.

D. Persistent Bookcase

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Recently in school Alina has learned what are the *persistent data structures*: they are data structures that always preserves the previous version of itself and access to it when it is modified.

After reaching home Alina decided to invent her own persistent data structure. Inventing didn't take long: there is a bookcase right behind her bed. Alina thinks that the bookcase is a good choice for a persistent data structure. Initially the bookcase is empty, thus there is no book at any position at any shelf.

The bookcase consists of n shelves, and each shelf has exactly m positions for books at it. Alina enumerates shelves by integers from 1 to n and positions at shelves — from 1 to m . Initially the bookcase is empty, thus there is no book at any position at any shelf in it.

Alina wrote down q operations, which will be consecutively applied to the bookcase. Each of the operations has one of four types:

- $1\ i\ j$ — Place a book at position j at shelf i if there is no book at it.
- $2\ i\ j$ — Remove the book from position j at shelf i if there is a book at it.
- $3\ i$ — Invert book placing at shelf i . This means that from every position at shelf i which has a book at it, the book should be removed, and at every position at shelf i which has not book at it, a book should be placed.
- $4\ k$ — Return the books in the bookcase in a state they were after applying k -th operation. In particular, $k = 0$ means that the bookcase should be in initial state, thus every book in the bookcase should be removed from its position.

After applying each of operation Alina is interested in the number of books in the bookcase. Alina got 'A' in the school and had no problem finding this values. Will you do so?

Input

The first line of the input contains three integers n , m and q ($1 \leq n, m \leq 10^3$, $1 \leq q \leq 10^5$) — the bookcase dimensions and the number of operations respectively.

The next q lines describes operations in chronological order — i -th of them describes i -th operation in one of the four formats described in the statement.

It is guaranteed that shelf indices and position indices are correct, and in each of fourth-type operation the number k corresponds to some operation before it or equals to 0 .

Output

For each operation, print the number of books in the bookcase after applying it in a separate line. The answers should be printed in chronological order.

Examples

input
2 3 3 1 1 1 3 2 4 0
output
1 4 0
input
4 2 6 3 2 2 2 2 3 3 3 2 2 2 2 3 2
output
2 1 3 3 2 4
input
2 2 2 3 2 2 2 1

output
2 1

Note

This image illustrates the second sample case.

E. Garlands

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Like all children, Alesha loves New Year celebration. During the celebration he and his whole family dress up the fir-tree. Like all children, Alesha likes to play with garlands — chains consisting of a lightbulbs.

Alesha uses a grid field sized $n \times m$ for playing. The rows of the field are numbered from 1 to n from the top to the bottom and columns are numbered from 1 to m from the left to the right.

Alesha has k garlands which he places at the field. He does so in the way such that each lightbulb of each garland lies in the center of some cell in the field, and each cell contains **at most one lightbulb**. Of course lightbulbs, which are neighbours in some garland, appears in cells neighbouring by a side.

The example of garland placing.

Each garland is turned off or turned on at any moment. If some garland is turned on then each of its lightbulbs is turned on, the same applies for garland turned off. Each lightbulb in the whole garland set is unique, and thus, being turned on, brings Alesha some pleasure, described by an integer value. Turned off lightbulbs don't bring Alesha any pleasure.

Alesha can turn garlands on and off and wants to know the sum of pleasure value which the lightbulbs, placed in the centers of the cells in some rectangular part of the field, bring him. Initially **all the garlands are turned on**.

Alesha is still very little and can't add big numbers. He extremely asks you to help him.

Input

The first line of the input contains three integers n , m and k ($1 \leq n, m, k \leq 2000$) — the number of field rows, the number of field columns and the number of garlands placed at the field respectively.

Next lines contains garlands set description in the following format:

The first line of a single garland description contains a single integer len ($1 \leq len \leq 2000$) — the number of lightbulbs in the garland.

Each of the next len lines contains three integers i, j and w ($1 \leq i \leq n, 1 \leq j \leq m, 1 \leq w \leq 10^9$) — the coordinates of the cell containing a lightbulb and pleasure value Alesha gets from it if it is turned on. The lightbulbs are given in the order they are forming a chain in the garland. It is guaranteed that neighbouring lightbulbs are placed in the cells neighbouring by a side.

The next line contains single integer q ($1 \leq q \leq 10^6$) — the number of events in Alesha's game. The next q lines describes events in chronological order. The i -th of them describes the i -th event in the one of the following formats:

- SWITCH i** — Alesha turns off i -th garland if it is turned on, or turns it on if it is turned off. It is guaranteed that $1 \leq i \leq k$.
- ASK $x_1 y_1 x_2 y_2$** — Alesha wants to know the sum of pleasure values the lightbulbs, placed in a rectangular part of the field. Top-left cell of a part has coordinates (x_1, y_1) and right-bottom cell has coordinates (x_2, y_2) . It is guaranteed that $1 \leq x_1 \leq x_2 \leq n$ and $1 \leq y_1 \leq y_2 \leq m$. There is **no more than 2000** events of this type in the input.

All the numbers in the input are integers.

Please note that the input is quite large, so be careful while using some input ways. In particular, it's not recommended to use `cin` in codes on C++ and class `Scanner` in codes on Java.

Output

For each **ASK** operation print the sum Alesha wants to know in a separate line. Print the answers in chronological order.

Examples

input
4 4 3 5 1 1 2 1 2 3 2 2 1 2 1 4 3 1 7 4 1 3 1 2 3 3 2 4 3 1 4 1 7 4 1 1 4 2 9 3 2 8 3 3 3 4 3 4 4 4 1 3 4 1

2 ASK 2 2 3 3 ASK 1 1 4 4
output
15 52
input
4 4 1 8 4 1 1 3 1 2 2 1 1 1 1 7 1 2 5 2 2 4 2 3 1 1 3 1 3 ASK 1 1 3 2 SWITCH 1 ASK 1 1 3 2
output
19 0

Note

This image illustrates the first sample case.