codeIt

A. Tournament

3 seconds, 256 megabytes

N teams competing in a knockout tournament. For simplicity, N is a power of two.

In the tournament's first round, the first team competes against the second, the 3rd against the 4th, and so on...

Given the number of teams participating in this tournament, you are required to answer Q queries.

Each query consists of two Integers T_1 and T_2 between 1 and N, representing identifiers of two different teams. Find the number of matches both teams need to play to face each other. If both teams will face each other directly in the first round, the answer is 1.

Input

The first line of the input consists of $2 \le N \le 2^{60}$ and $1 \le Q \le 10^5$ the number of the teams, and the number of queries respectively.

Then Q Lines follow, each consists of two teams $1 \le T_1 \ne T_2 \le N$.

Output

For each query, report a single integer, the number of rounds until the two teams will face other if they win every single round before that.

input

```
4 6
1 2
1 3
1 4
2 3
2 4
3 4
```

output

```
1
2
2
2
2
1
```

B. Train Trips

4 seconds, 256 megabytes

Traveling by train is always the best way to discover a country's rivers, landscapes, and forests.

In this problem, we consider an infinite rail that has n stations where the distance between two successive stations is $(dist(station_i, station_{i+1}) = 1)$. If you want to travel from a starting station x to a destination station y ($y \ne x$), the condition $|x - y| < d$ should be satisfied where d is a distance given as input.

To go from a given starting station, there are multiple paths that you can use. In this problem, you are asked to count the total number of possible paths for each starting station i ($1 \le i \le n$) while changing stations in the way exactly k times (not necessarily k distinct stations).

Two paths are considered different if j^{th} station ($1 \le j \le k+1$) in a path is different than j^{th} station in another path. For example, the path `1 2 1 3` is different than the path `1 3 1 2`.

Input

The first line contains an integer T ($1 \le T \le 5$) denoting the number of test cases.

The next T lines contain three integers n, d et k ($1 \le n, d, k \le 3000$) described in the statement.

Output

For each line, print n integers denoting the number of unique paths for each starting station s ($1 \le s \le n$). The i^{th} number ($1 \le i \le n$) in each line corresponds to the number of possible paths from i^{th} station for the corresponding test case.

Since the numbers might be very large, print all answers modulo $10^9 + 7$.

input

```
2
5 1 4
5 2 1
```

output

```
0 0 0 0 0
1 2 2 2 1
```

C. 24/7 Cafe

1 second, 256 megabytes

Diaa runs a 24/7 fast food cafe. This cafe contains M tables, Each table has a limited number of people that it can accommodate, He magically learned that N groups of people will visit his cafe. For each group, we know the arrival time, The leaving time if they find an empty table, And the size of the group(number of people in the group).

The cafe spends no time serving each client, but if a group comes in and sees that there are no tables that can hold them all, Then the group doesn't want to wait and leaves the cafe immediately, Otherwise the waiter will take the group to the smallest table that can accommodate them.

If a group leaves at the i_th minute, Then the table they were using will be available starting from the next minute($(i+1)\_th$ minute). Each table can only be used by one group at a time.

What is the number of groups the cafe will serve?

Input

The first line contains two integers n and m ($1 \le N, M \le 10^5$), That is the number of groups, And the number of tables respectively.

Each of the following n lines has three space-separated integers L, R and S ($1 \le L < R \le 10^9, 1 \le S \le 10^9$), representing the time when the i-th group comes into the cafe, The time when this group leaves and the size of the group respectively.

t's guaranteed that no two groups arrive at the same time (all arriving times are distinct). The next line contains M space-separated integers $a\_1, a\_2, \ldots, a\_n$, Representing the size of each table.

Output

Print one integer — The number of groups the cafe will serve.

input

```
3 3
1 2 3
3 4 5
5 6 1
1 4 3
```

output

```
2
```

D. Magical Frog

1 second, 256 megabytes

Reda started this year studying at HogwarIT for magic and technology, his Potions professor insisted that all the students get magical pets to help them with the rest of the course, Reda picked a Frog.

For the first homework, students are required to train their pets to do some math tricks. Reda chose to observe his pet's movement and tried to find some patterns to help him find a suitable trick to teach it.

The frog lived in the HogwarIT dormitory staircase, each day The frog starts at the bottom of the stairs (i.e, stair number 0) and can jump to any of the next K stairs, (i.e if frog is at stair i it can hump to stair $i + j$ where $(i + j) \leq N$ and $1 \leq j \leq K$) and it keeps doing jumps until it reaches the N'th stair.

Can you help Reda count the number of ways the frog can reach the N'th stair since this number can be too big output it modulo $10^9 + 7$.

Two ways are considered different if the sequence of jumps taken in both of them is different.

Input

The input contains a single line with 2 integers $1 \leq N \leq 10^9$ $1 \leq K \leq 20$ As described above. Also it is guaranteed that $K \leq N$.

Output

Print one single line, the number of different ways to reach the N'th stair.

input

```
3 2
```

output

```
3
```

input

```
6 3
```

output

```
24
```

input

```
2 2
```

output

```
2
```

E. Youtube Channels

1 second, 256 megabytes

We have n youtube channels (numbered from 1 to n). Then we have q queries.

Each query will be one of the following types.

- Query type 1 (format 1 x y): x members followed channel y.
- Query type 2 (format 2 x y): x members unfollowed channel y.
- Query type 3 (format 3): Print the channel with the maximum number of followers
- Query type 4 (format 4): Print the channel with the minimum number of followers.

The number of followers will never go below 0 after an unfollow query.

For queries of type 3 and 4, if there are multiple channels with the same number of followers, print the one with the smallest ID.

Input

The first line of the input contains two integers $1 \leq n \leq 10^5$ and $1 \leq q \leq 10^5$ denoting respectively the number of channels and the number of queries.

q lines follow each contains a query in one of the formats described in the statement.

For Queries of type 1 and 2, we have $0 \leq x \leq 10^7$ and $1 \leq y \leq n$.

Output

For each query of type 3 and 4 print the corresponding answer.

input

```
4 12
1 5 1
3
4
1 6 2
3
4
1 4 3
1 5 4
2 2 1
2 3 2
3
4
```

output

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

F. Dehbi and his Friends

15 seconds, 256 megabytes

Dehbi and his friends are playing a game.

The game is described as follows: She has x friends who are all located in different starting points, and they want to reach a destination while Dehbi is waiting for them.

Dehbi wants your help. She wants to know the first time while at least one of his x friends reaches him.

Input

The Input starts with a line containing a single integer T, the number of test cases.

Each test case will starts with two integers n and m that denotes the number of locations where Dehbi's friends stand and the number of roads respectively, $(2 \leq n \leq 10^5)$, and $(1 \leq m \leq 10^5)$.

Each of the next m lines will contain three integers u, v and w that represent a road (u, v) and its length, where $1 \leq w \leq 10^3$ and $1 \leq u, v \leq n$.

Next line contains x that represents the number of Dehbi's friends, $(1 \le x \le n)$. Next line contains x numbers that represent the initial starting locations of Dehbi's friends.

Last line will contain a single integer which is the final destination where Dehbi is waiting for his friends.

Output

If no one can reach him output -1, otherwise output the first time when one of her x friends will get to his place. In other words, the time needed by the fastest friend to reach him.

input
```
1
3 2
1 3 4
2 3 10
2
1 2
3
```
output
```
4
```

G. Tree Permutations

1 second, 256 megabytes

The algorithm below generates a random permutation by randomly walking in a tree T.

`permutation = [] DFS(T, root): permutation.add(root) ForAll` $v \in$ `Shuffle(Children(root)) DFS(T, v)`

Let $\delta(r)$ be the number of permutations that can be generated by the algorithm above if the tree is rooted at r. Your task is to calculate the expected value of $\delta(r)$ if every vertex from the tree can be chosen with equal probability.

Input

The first line contains a single integer $2 \le n \le 10^5$, denoting the number of nodes in the tree. The following $n-1$ lines contains each two integers u, and v denoting an edge in the tree T. ($u \ne v$), and node indexes are between 1 and n.

Output

Output a single integer denoting the expected number of permutations of the tree.

The expected value can be very large, hence if the answer is $\frac{a}{b}$, then you should output $a \times b^{-1} \backslash tmodulo 10^9 + 7$, where b^{-1} is the multiplicative inverse of b (the inverse is guaranteed to exist).

input
```
2
1 2
```
output
```
1
```

input
```
3
1 2
1 3
```
output
```
333333337
```

input
```
4
1 2
1 3
1 4
```
output
```
3
```

The answer to the second test case is:

$$E = \frac{\delta(1)+\delta(2)+\delta(3)}{3} = \frac{2+1+1}{3} = 4 \times 3^{-1} \backslash tmodulo 10^9 + 7 = 4$$

The answer to the third test case is:

$$E = \frac{\delta(1)+\delta(2)+\delta(3)+\delta(4)}{4} = \frac{6+2+2+2}{4} = \frac{12}{4} = 3$$

H. Cutting with Regexes

1 second, 256 megabytes

Given a list of regular expressions `R`, and a text `T`. Calculate the number of ways you can cut `T` such that every cut can be matched with one of the regular expressions from `R`.

For example if we have 3 regular expressions `aa, b, aabb`, and the the text `aabb`. Then there are two possible ways of cutting the input text. These are:

- `aa|b|b`, and
- `aabb`

Input

The first line contains a single integer n, the number of regular expressions. Next (n) lines contain each one regex. The total size of all regular expressions is less than or equal to 2000. The last line of the input contains the text (T), where $1 \le |\backslash tT| \le 2000$.

Output

Output the number of ways you can cut the (T). Since that number can be large, output it modulo $10^9 + 7$.

input
```
3
aa
b
aabb
aabb
```
output
```
2
```

input
```
1
a+
aaaaa
```
output
```
16
```

The regular expression follow the rules below:

- Regular expressions' alphabet is lowercase english letters.
- The modifier modifier `'+'` can optionally follow a letter, and means 1 or more of that letter. For example `a+` means one or more `a`s.
- All regular expressions are unique, and no two regular expressions can match the same word.

- Finally, all regular expressions are simplified, meaning that you cannot get something like `a+a` because it can be simplified to `a+` only.

I. The ENIAC

3 seconds, 256 megabytes

Kathleen Antonelli was a female computer programmer, and one of the main programmers of the ENIAC – the first electronic digital computer in history.

One day, Kathleen passed by the ENIAC while it was open. She noticed there was a complicated electric network inside the ENIAC. The network can be seen as a directed graph with edges connecting vertices.

Kathleen was a very curious woman. She wanted to know whether, for a given pair of vertices, there is a way to start from one of them, follow some sequence of directed edges and arrive to the other one. Can you help answer her questions and cure her curiosity?

Input

The input consists of multiple test cases.

The first line of each test case contains three integers n, m, and q ($2 \le n \le 300$, $1 \le m \le 1000$, $1 \le q \le 1000$) denoting the number of vertices, number of edges, and number of questions, respectively. Vertices are numbered from 1 to n.

The following m lines describe the graph with one edge per line. Each line contains two integers a and b ($a \ne b$, $1 \le a, b \le n$) meaning there is an edge going from vertex a to vertex b (and not the other way around). The pair (a, b) may appear more than once.

The following q lines contain Kathleen's questions, with one question per line. Each question contains two integers u and v ($1 \le u, v \le n$), asking whether it is possible to start from vertex u and follow a sequence of directed edges to arrive to vertex v.

Output

For every question, print a separate line with one word. Print "`YES`" (without quotes) if it is possible to start from vertex u and follow a sequence of directed edges to arrive to vertex v. Otherwise, print "`NO`" (without quotes).

input
```
3 2 3
1 2
2 3
1 3
3 2
1 2
```

output
```
YES
NO
YES
```

input
```
3 2 2
1 2
2 1
1 1
3 3
```

output
```
YES
NO
```

J. P vs NP

2 seconds, 128 megabytes

In order to get the famous green Accepted notification for this problem, you'll have to solve the P vs NP problem (Or skip if you can't!).

Seriously... you want to take the challenge? Ok then. A problem is said to be 'NP' if the number of vowels in its name is twice the number of words, otherwise, it's called a 'P' problem.

You will be given a list of names and we want you to say if the problem is P or NP.

You still think this is easy?

Input

The input consists of multiple lines, each line contains a string *Name* denoting the name of a problem. Each name consists of multiple lowercase strings separated by a single space ($|Name| \le 10^6$).

Output

For each line print "NP" if the corresponding problem is NP or "P" otherwise (quotes for clarity).

input
```
azuz the astronaut
sabrina luv
```

output
```
P
NP
```

Note that vowels are {'a', 'e', 'i', 'o', 'u', 'y'}.