# A. The Monster

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

As Will is stuck in the Upside Down, he can still communicate with his mom, Joyce, through the Christmas lights (he can turn them on and off with his mind). He can't directly tell his mom where he is, because the monster that took him to the Upside Down will know and relocate him.


Thus, he came up with a puzzle to tell his mom his coordinates. His coordinates are the answer to the following problem.

A string consisting only of parentheses ('`(`' and '`)`') is called a bracket sequence. Some bracket sequence are called correct bracket sequences. More formally:

- Empty string is a correct bracket sequence.
- if $s$ is a correct bracket sequence, then $(s)$ is also a correct bracket sequence.
- if $s$ and $t$ are correct bracket sequences, then $st$ (concatenation of $s$ and $t$) is also a correct bracket sequence.

A string consisting of parentheses and question marks ('`?`') is called pretty if and only if there's a way to replace each question mark with either '`(`' or '`)`' such that the resulting string is a **non-empty** correct bracket sequence.

Will gave his mom a string $s$ consisting of parentheses and question marks (using Morse code through the lights) and his coordinates are the number of pairs of integers $(l, r)$ such that $1 \leq l \leq r \leq |s|$ and the string $s_l s_{l+1} \dots s_r$ is pretty, where $s_i$ is $i$-th character of $s$.

Joyce doesn't know anything about bracket sequences, so she asked for your help.

### Input

The first and only line of input contains string $s$, consisting only of characters '`(`', '`)`' and '`?`' $(2 \leq |s| \leq 5000)$.

### Output

Print the answer to Will's puzzle in the first and only line of output.

#### Examples

input
```
((?))
```
output
```
4
```

input
```
??()??
```
output
```
7
```

### Note

For the first sample testcase, the pretty substrings of $s$ are:

1. "`(?`" which can be transformed to "`()`".
2. "`?)`" which can be transformed to "`()`".
3. "`((?)`" which can be transformed to "`(())`".
4. "`(?))`" which can be transformed to "`(())`".

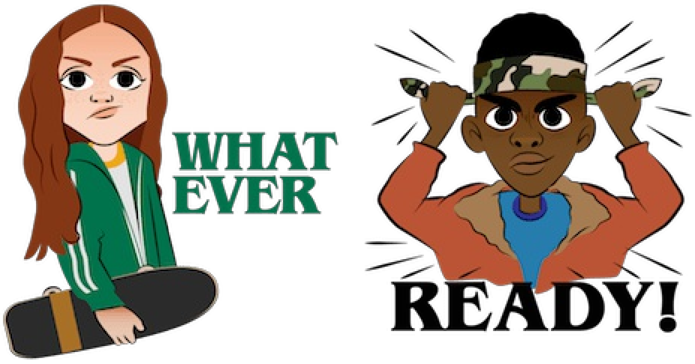
For the second sample testcase, the pretty substrings of $s$ are:

1. "`??`" which can be transformed to "`()`".
2. "`()`".
3. "`??()`" which can be transformed to "`()()`".
4. "`?()?`" which can be transformed to "`(())`".
5. "`??`" which can be transformed to "`()`".
6. "`()??`" which can be transformed to "`()()`".
7. "`??()??`" which can be transformed to "`()()()`".

# B. MADMAX

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

As we all know, Max is the best video game player among her friends. Her friends were so jealous of hers, that they created an actual game just to prove that she's not the best at games. The game is played on a directed acyclic graph (a DAG) with $n$ vertices and $m$ edges. There's a character written on each edge, a lowercase English letter.




Max and Lucas are playing the game. Max goes first, then Lucas, then Max again and so on. Each player has a marble, initially located at some vertex. Each player in his/her turn should move his/her marble along some edge (a player can move the marble from vertex $v$ to vertex $u$ if there's an outgoing edge from $v$ to $u$). If the player moves his/her marble from vertex $v$ to vertex $u$, the "character" of that round is the character written on the edge from $v$ to $u$. There's one additional rule; the ASCII code of character of round $i$ should be **greater than or equal** to the ASCII code of character of round $i$ - $1$ (for $i > 1$). The rounds are numbered for both players together, i. e. Max goes in odd numbers, Lucas goes in even numbers. The player that can't make a move loses the game. The marbles may be at the same vertex at the same time.

Since the game could take a while and Lucas and Max have to focus on finding Dart, they don't have time to play. So they asked you, if they both play optimally, who wins the game?

You have to determine the winner of the game for all initial positions of the marbles.

### Input

The first line of input contains two integers $n$ and $m$ ($2 \le n \le 100$, $1 \le m \le \frac{n(n-1)}{2}$).

The next $m$ lines contain the edges. Each line contains two integers $v$, $u$ and a lowercase English letter $c$, meaning there's an edge from $v$ to $u$ written $c$ on it ($1 \le v, u \le n$, $v \ne u$). There's at most one edge between any pair of vertices. It is guaranteed that the graph is acyclic.

### Output

Print $n$ lines, a string of length $n$ in each one. The $j$-th character in $i$-th line should be 'A' if Max will win the game in case her marble is initially at vertex $i$ and Lucas's marble is initially at vertex $j$, and 'B' otherwise.

#### Examples

input

```
4 4
1 2 b
1 3 a
2 4 c
3 4 b
```

output

```
BAAA
ABAA
BBBA
BBBB
```

input

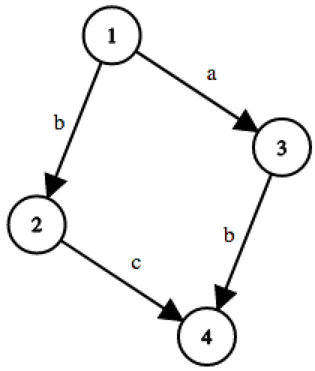
```
5 8
5 3 h
1 2 c
3 1 c
3 2 r
5 1 r
4 3 z
5 4 r
5 2 h
```

output

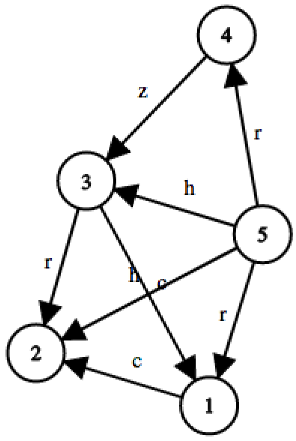
```
BABBB
BBBBB
AABBB
AAABA
AAAAB
```

### Note

Here's the graph in the first sample test case:




Here's the graph in the second sample test case:



# C. Pollywog

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

As we all know, Dart is some kind of creature from Upside Down world. For simplicity, we call their kind *pollywogs*. Dart and $x - 1$ other pollywogs are playing a game. There are $n$ stones in a row, numbered from $1$ through $n$ from left to right. At most $1$ pollywog may be sitting on each stone at a time. Initially, the pollywogs are sitting on the first $x$ stones (one pollywog on each stone).


Dart and his friends want to end up on the last $x$ stones. At each second, the leftmost pollywog should jump to the right. A pollywog can jump at most $k$ stones; more specifically, a pollywog can jump from stone number $i$ to stones $i + 1, i + 2, \dots i + k$. A pollywog can't jump on an occupied stone. Jumping a distance $i$ takes $c_i$ amounts of energy from the pollywog.

Also, $q$ stones are *special* Each time landing on a special stone $p$, takes $w_p$ amounts of energy (in addition to the energy for jump) from the pollywog. $w_p$ could be negative, in this case, it means the pollywog absorbs $|w_p|$ amounts of energy.

Pollywogs want to spend as little energy as possible (this value could be negative).

They're just pollywogs, so they asked for your help. Tell them the total change in their energy, in case they move optimally.

### Input

The first line of input contains four integers, $x$, $k$, $n$ and $q$ ($1 \le x \le k \le 8$, $k \le n \le 10^8$, $0 \le q \le min(25, n - x)$) — the number of pollywogs, the maximum length of jump, the number of stones and the number of special stones.

The next line contains $k$ integers, $c_1, c_2, \dots c_k$, separated by spaces ($1 \le c_i \le 10^9$) — the energetic costs of jumps.

The next $q$ lines contain description of the special stones. Each line contains two integers $p$ and $w_p$ ($x + 1 \le p \le n$, $|w_p| \le 10^9$). All $p$ are distinct.

### Output

Print the minimum amount of energy they need, in the first and only line of output.

#### Examples

input

```
2 3 10 2
1 2 3
5 -10
6 1000
```

output

```
6
```

input

```
4 7 85 3
17 5 28 4 52 46 6
59 -76
33 -69
19 2018
```

output

```
135
```

# D. Stranger Trees

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

Will shares a psychic connection with the Upside Down Monster, so everything the monster knows, Will knows. Suddenly, he started drawing, page after page, non-stop. Joyce, his mom, and Chief Hopper put the drawings together, and they realized, it's a labeled tree!


A tree is a connected acyclic graph. Will's tree has $n$ vertices. Joyce and Hopper don't know what that means, so they're investigating this tree and similar trees. For each $k$ such that $0 \le k \le n - 1$, they're going to investigate all labeled trees with $n$ vertices that share exactly $k$ edges with Will's tree. Two labeled trees are different if and only if there's a pair of vertices $(v, u)$ such that there's an edge between $v$ and $u$ in one tree and not in the other one.

Hopper and Joyce want to know how much work they have to do, so they asked you to tell them the number of labeled trees with $n$ vertices that share exactly $k$ edges with Will's tree, for each $k$. The answer could be very large, so they only asked you to tell them the answers modulo $1000000007 = 10^9 + 7$.

### Input

The first line of input contains a single integer $n$ $(2 \le n \le 100)$ — the size of the tree.

The next $n - 1$ lines contain the edges of Will's tree. Each line contains two integers $v$ and $u$ $(1 \le v, u \le n, v \ne u)$, endpoints of an edge. It is guaranteed that the given graph is a tree.

### Output

Print $n$ integers in one line. $i$-th integer should be the number of the number of labeled trees with $n$ vertices that share exactly $i - 1$ edges with Will's tree, modulo $1000\,000\,007 = 10^9 + 7$.

#### Examples

input

```
3
1 2
1 3
```

output

```
0 2 1
```

input

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

output

```
1 7 7 1
```

input

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

output

```
0 9 6 1
```

# E. Upside Down

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

As we all know, Eleven has special abilities. Thus, Hopper convinced her to close the gate to the Upside Down World with her mind. Upside down monsters like to move between the worlds, so they are going to attack Hopper and Eleven in order to make them stop. The monsters live in the vines. The vines form a tree with $n$ vertices, numbered from $1$ through $n$. There's a lowercase English letter written in each tunnel (edge).

Upside down is a magical world. There are $m$ types of monsters in upside down, numbered from $1$ through $m$. Each type of monster has a special word that gives them powers. The special word of type $i$ is $s_i$. There are $q$ monsters in upside down. Each one is at a junction (vertex) and is going to some other junction. If monster of type $k$ goes from junction $i$ to junction $j$, the power it gains is the number of times it sees its special world ($s_k$) consecutively in the tunnels. More formally:

If $f(i, j)$ is the string we get when we concatenate the letters written in the tunnels on the shortest path from $i$ to $j$, then the power the monster gains is the number of occurrences of $s_k$ in $f(i, j)$.

Hopper and Eleven want to get prepared, so for each monster, they want to know the power the monster gains after moving.

### Input

The first line of input contains three integers, $n$, $m$ and $q$ ($2 \le n \le 10^5$, $1 \le m, q \le 10^5$).

The next $n$ - $1$ lines contain the tunnels (edges). Each line contains two integers $v$ and $u$ ($1 \le v, u \le n$, $v \ne u$) and a lowercase English letter $c$, meaning there's a tunnel connecting junctions $v$ and $u$ written $c$ in it. It is guaranteed that the given graph is a tree.

The next $m$ lines contain the special words. $i$-th line of them contains a single string $s_i$ ($1 \le |s_i| \le 10^5$), consisting of lowercase English letters. It is guaranteed that $|s_1| + |s_2| + \ldots + |s_m| \le 10^5$).

The next $q$ lines contain the monsters. Each line contains three integers $i$, $j$ and $k$ ($1 \le i, j \le n$, $i \ne j$, $1 \le k \le m$), meaning a monster of type $k$ is going from junction number $i$ to junction number $j$.

### Output

Print $q$ lines. $i$-th line should contain a single integer, the power the $i$-th monster gains after moving.

#### Examples

**input**

```
6 4 5
1 6 b
2 3 a
1 2 b
5 3 b
4 5 b
a
b
bb
aa
1 2 1
6 2 3
1 6 2
4 5 4
1 6 2
```

**output**

```
0
1
1
0
1
```

**input**

```
10 6 7
1 3 s
10 1 d
```

```
2 6 s
5 2 d
7 4 l
8 9 d
8 10 l
7 2 d
8 7 l
dl
dssld
d
d
l
sl
4 5 4
3 7 5
10 6 2
3 1 4
7 5 6
10 9 4
9 8 4
```

**output**

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