Codeforces
Sponsored by Telegram

Codeforces Round #459 (Div. 2)

A. Eleven

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

Eleven wants to choose a new name for herself. As a bunch of geeks, her friends suggested an algorithm to choose a name for her. Eleven wants her name to have exactly n characters.

Her friend suggested that her name should only consist of uppercase and lowercase letters 'O'. More precisely, they suggested that the i-th letter of her name should be 'O' (uppercase) if i is a member of Fibonacci sequence, and 'o' (lowercase) otherwise. The letters in the name are numbered from 1 to n. Fibonacci sequence is the sequence f where

- $f_1 = 1$,
- $f_2 = 1$,
- $f_n = f_{n-2} + f_{n-1}$ $(n > 2)$.

As her friends are too young to know what Fibonacci sequence is, they asked you to help Eleven determine her new name.

Input

The first and only line of input contains an integer n $(1 \le n \le 1000)$.

Output

Print Eleven's new name on the first and only line of output.

Examples

input
```
8
```
output
```
OOOoOooO
```

input
```
15
```
output
```
OOOoOooOooooOoo
```

B. Radio Station

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

As the guys fried the radio station facilities, the school principal gave them tasks as a punishment. Dustin's task was to add comments to nginx configuration for school's website. The school has n servers. Each server has a name and an ip (names aren't necessarily unique, but ips are). Dustin knows the ip and name of each server. For simplicity, we'll assume that an nginx command is of form "`command ip;`" where `command` is a string consisting of English lowercase letter only, and `ip` is the ip of one of school servers.

Each ip is of form "`a.b.c.d`" where a, b, c and d are non-negative integers less than or equal to 255 (with no leading zeros). The nginx configuration file Dustin has to add comments to has m commands. Nobody ever memorizes the ips of servers, so to understand the configuration better, Dustin has to comment the name of server that the ip belongs to at the end of each line (after each command). More formally, if a line is "`command ip;`" Dustin has to replace it with "`command ip; #name`" where `name` is the name of the server with ip equal to `ip`.

Dustin doesn't know anything about nginx, so he panicked again and his friends asked you to do his task for him.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 1000$).

The next n lines contain the names and ips of the servers. Each line contains a string `name`, name of the server and a string `ip`, ip of the server, separated by space ($1 \le |name| \le 10$, $name$ only consists of English lowercase letters). It is guaranteed that all ip are distinct.

The next m lines contain the commands in the configuration file. Each line is of form "`command ip;`" ($1 \le |command| \le 10$, `command` only consists of English lowercase letters). It is guaranteed that ip belongs to one of the n school servers.

Output

Print m lines, the commands in the configuration file after Dustin did his task.

Examples

input

```
2 2
main 192.168.0.2
replica 192.168.0.1
block 192.168.0.1;
proxy 192.168.0.2;
```

output

```
block 192.168.0.1; #replica
proxy 192.168.0.2; #main
```

input

```
3 5
google 8.8.8.8
codeforces 212.193.33.27
server 138.197.64.57
redirect 138.197.64.57;
block 8.8.8.8;
cf 212.193.33.27;
unblock 8.8.8.8;
check 138.197.64.57;
```

output

```
redirect 138.197.64.57; #server
block 8.8.8.8; #google
cf 212.193.33.27; #codeforces
unblock 8.8.8.8; #google
check 138.197.64.57; #server
```

C. 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:

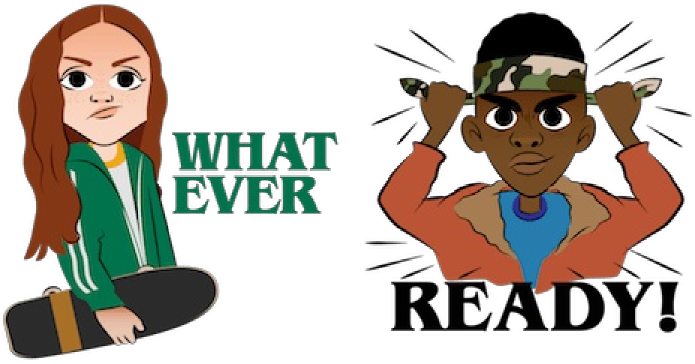
1. "??" which can be transformed to "()".
2. "()".
3. "??()" which can be transformed to "()()".
4. "?()?" which can be transformed to "(())".
5. "??" which can be transformed to "()".

6. "`()??`" which can be transformed to "`()()`".
7. "`??()??`" which can be transformed to "`()()()`".

D. 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

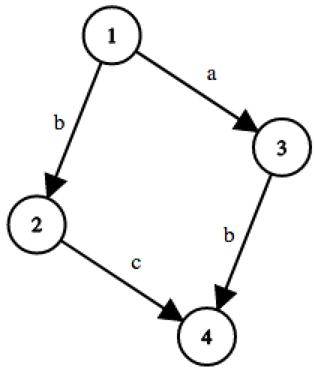
```
5 8
5 3 h
1 2 c
3 1 c
3 2 r
5 1 r
4 3 z
5 4 r
5 2 h
```

output

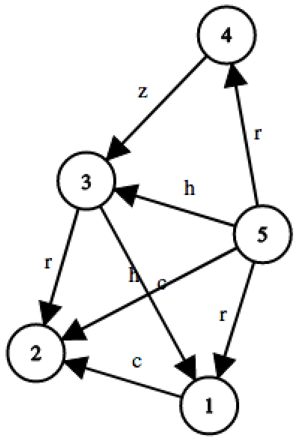
```
BABBB
BBBBB
AABBB
AAABA
AAAAB
```

Note

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



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



E. 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
```