Problem A. Artifacts

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

In the new game the hero have N kinds of energy, labeled by sequential integers between 1 to N. Before set off for an adventure, hero can charge some of his energy, which allow him to set the amount a_i of every kind of energy a non-negative integer arbitrarily.

Different kind of energy is set independently.

When different energies mix, they become unstable. If there exist two integers u and v ($u \leq v$), where a_u, $a_u + 1$, …, a_v are positive integers with sum more than K, the hero is destroyed and game is lost.

During the adventure, the story may develop differently depending on the amount of a certain kind of energy. Sometimes there can appear the artifacts, but certain artifacts may only appear in some certain sub-story.

The scenario of the game is given as some pseudocode.

```
<scenario> := game ( <constant>, <constant> ) { <body> }
<body> := <statement> <body> | empty
<statement> := <branch-statement> | <artifact>; | { <body> }
<branch-statement> := if ( a [ <constant> ] >= <constant> ) <statement>
                      [else <statement>]
<artifact> := art ( <constant> )
```

Here `<constant>` denotes non-negative integers.

Lets explain the formal definition:

The scenario is presented as a function `game(n,k`, where n is number of energy type available, and K is maximal safe for the hero sum. Three types of `<statement>` are possible:

1. `art (id)` — artifact number x is generated;
2. `if (a[x]>=y) <statement> [else <statement>]` — similar to the branch statement in C-based languages; `else` is matched with the nearest `if`.
3. { `<statements >`} — a block containing zero or more several statements.

Note that the tokens may be separated with arbitrary number of blanks, including the spaces and line feeds.

Your goal is given the scenario, determine how many artifacts are collectable.

Input

A correct scenario with length no more than 3 mebibytes. It is guaranteed that $1 \leq n \leq 10^5$, argument of `art()` function id is non-negative integer between 0 and $2^{20} - 1$, and in `if` statement $1 \leq x \leq n$ and $0 \leq y \leq 10^9$.

Output

Print one integer — number of collectible artifacts

Example

standard input	standard output

game (3,1) {
art(0);
if (a[1] >= 1)
art (1);
else if (a[1] >= 2) art(2);
if (a[2] >= 1)
if (a[3] >= 1) art(3);
else art(4);
{art(0)};
}
	3

Note

Artifacts 0, 1 and 4 may be collected.

Problem B. Brackets and Dots

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 mebibytes

Andrey likes strings, composed from the brackets, and he likes to replace brackets with dots...

Andrey got the string of length N, and applies M operations to it. Each operation is given by two integers l_i and r_i, and in the result of thos operation in substring $s_{l_i}s)l_i + 1, \ldots, s_{r_i}$ longest regular bracket subsequence is replaced by dots such as each bracket is replaced by an one dot.

If the substring have several longest regular bracket subsequences, then the minimal one is selected. Regular bracket sequences are compared in next way: lets a_1, a_2, $\ldots$, a_l — indices of the opening brackets for the first subsequence, and c_1, c_2, $\ldots$, c_n — indices of the opening brackets for the second subsequence. If for some k $a_i = c_i$ $(1 \le i < k)$ and $a_k > c_k$, then first subsequence is considered lesser, than second one. If the opening brackets are at the same places in both sequences, then consider indices of the closing brackets b_1, b_2, $\ldots$, b_l for the first subsequence and d_1, d_2, $\ldots$, d_n for the second subsequence. If exists k $(1 \le k \le l)$ such as $b_i = d_i$ $(1 \le i < k)$ and $b_k < d_k$, then first subsequence is considered lesser.

For each operation print number of brackets replaced by dots.

Input

First line of the input contains one non-empty string S ($|S| \le 5 \cdot 10^5$), consisting of opening and closing brackets (‘`(`’ and ‘`)`’).

Second line contains one integer M $(1 \le M \le 5 \cdot 10^5)$ — number of operations.

Each of the next M lines describe one operation and consists of two integers l_i and r_i $(1 \le l_i \le r_i \le N)$ — starting and ending indices of the substring, respectively.

Output

For each operation print one integer — number of brackets replaced by dots.

Example

standard input	standard output

`((())()))`
`3`
`4 9`
`2 8`
`1 10`
	`4`
	`2`
	`4`

Problem C. Crossword

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

While creating the crosswords often appear next subtask:

Given four words, create the crossword-rectangle from them, i.e:

- Two words must be placed horizontally, and two — vertically.
- Horizontally placed words must be read from left to right, vertically placed words — upside down.
- Each word must have intersection exactly with two other words; each horizontally placed word intersects with both vertically placed words and each vertically placed word intersects with both horizontally placed words.
- Area of the empty rectangle inside the words must be greater than zero (i.e. no intersection can use neighbour letters).

You have four different words. Find the number of different ways to create the crossword-rectangle from them.

Input

Input consists of 4 lines, each containing non-empty word consisting of at least 3 and at most 100 lowercase English letters. Words are pairwise distinct.

Output

Print one integer — number of different ways to create the crossword-rectangle from the given words.

Example

standard input	standard output
aaa axa aya aza	24
aaaa abba baab bbbb	40

different

Problem D. Digit

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Lets write on the paper decimal notation for all integers between 1 and N. Find the most frequent decimal digit on the paper. If there are more than one such digit, choose maximal one.

Input

First line of the input contains one integer N ($1 \le N \le 10^{100\,000}$).

Output

Print one digit — answer to the problem.

Example

standard input	standard output
100	1
99	9

Problem E. Enormous Table

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given the infinite table:

```
1  2  6  7  15 ...
3  5  8  14 ...
4  9  13 ...
10 12 ...
11 ...
```

Print the value of the element in a-th row and b-th column.

Input

Input contains two integers a and b — row and column ($1 \leq a, b \leq 10^9$).

Output

Print one integer — answer to the problem.

Example

standard input	standard output
2 3	8

Problem F. Funny Language

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

The Martian language consists of M words; each word is sequence of the Martian letters (at Earth they are represented as integers between 1 and $8 \cdot 10^4$).

The sentence in Martian language is concatenation of the words used in it. Alice arranges the sentences of M pairwise distinct Martian words. For each sentence, she calculates the sum of great common divisors of all different intervals of the sentence and writes it down.

When all $M!$ possible sentences are built, Alice sums all the written values and calculates the answer modulo $10^9 + 7$.

Given the dictionary of Martian words, help Alice to find the answer.

Input

The first line of the input contains one integer M ($1 \le M \le 8 \cdot 10^4$) — number of words in the Martian language. Then M lines follow. Each line start with a integer S — length of the word ($1 \le S \le 8 \cdot 10^4$), then S integers between 1 and $8 \cdot 10^4$ follow — body of the word.

The sum of S does not exceed $8 \cdot 10^4$.

Output

Print the answer modulo $10^9 + 7$.

Example

standard input	standard output

2
2 1 2
1 1
	14

1 3 1 1 1	6

Problem G. Game of Tic-Tac-Toe

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

This is interactive problem

Your are to play against the jury program in classic tic-tac-toe.

Tic-tat-toe is a game for two players, X and O, who take turns marking the spaces in a 3×3 grid. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row wins the game. If the grid is filled and noone won, the game considered to be a draw.

Jury program plays with O, you use X. If you lose the game, you fail.

Interaction Protocol

First jury program tells who plays first: 'X' if you begin the game or 'O' (uppercase English O) otherwise.

Then interaction starts.

It its jury program' turn now, it tells you coordinates of his turn as pair of integers r and c, denoting number of the row and the column; rows are enumerated by sequential integers from 1 to 3 upside down, columns are enumerated by sequential integers from 1 to 3 from left to right. Each turn is printed in the new line, integers are separated by the one space. You may assume that selected cell is not busy.

At your turn you must print your turn in the same format. If you make an incorrect turn, you immediately fail the test.

When game ends, you receive a message from the jury program: "WIN" if you won, "LOSE" if you lost and "DRAW" if game ends up with a draw.

standard input	standard output

X
1 3
2 3
WIN
	1 1
	2 1
	3 1

O
1 1
3 1
2 3
1 2
2 2
DRAW
	
	3 3
	1 3
	2 1
	3 2

Note

Please end each output operation by end-of-line and do not forget to flush output buffer. Here are some examples of flush functions:

- For Pascal: **flush(output)**;
- For /++: **fflush(stdout)** or **cout.flush()**;
- For Java: **System.out.flush()**;
- For Python: **sys.stdout.flush()** from library **sys**;
- For C#: **Console.Out.Flush()**;

Problem H. Hill and Subhill

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Lets define *hill* as a set of points

$$P(n) = \{(x, y, z) | 1 \le z \le y \le x \le n\}$$

and *subhill* as a set of points

$$sP(x, y, z, a) = \{(x + i, y + j, z + k) | 0 \le k \le j \le i < a\}$$

You are given a hill $P(N)$, every integer point of $P(N)$ is assigned a value 0. Then you must perform M operations $A(x_i, y_i, z_i, a_i)$ meaning that you should add 1 to value of every integer point in $sP(x_i, y_i, z_i, a_i)$. Then you must answer Q queries $Q(x_j, y_j, z_j, a_j)$ requesting the sum of values of integer points in $sP(x_j, y_j, z_j, a_j)$. It is guaranteed that sP is subset of P.

Input

First line of the input contains three integers N, M and Q ($1 \le N \le 100$, $1 \le M \le 10^5$, $1 \le Q \le 10^5$). Then M lines follow, describing the operations in order they are performed. Each line contain four integers x_i, y_i, z_i, a_i Then Q lines follow, describing the queries in order they are must be answered. Each line contain four integers x_j, y_j, z_j, a_j. It is guaranteed that parameters of the queries and operations are correct.

Output

For each query, output a line with one integer — the answer to the query.

Example

standard input	standard output

2 1 2
1 1 1 2
1 1 1 2
2 1 1 1
	4
	1

Problem I. It is panic?

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Your friend works in 911 service. One of his duties is to define if the SMS message is panic. The SMS message is considered as panic if it have form “AAA...!!!!”, i.e. it can be split onto two non-empty parts such as left part consists only of uppercase ‘A’, and right part consists only of ‘!’ signs.

Given a message, check if it is panic.

Input

Input consists of nonempty string composed of upper- and lower case English letters and signs ‘!’ and ‘?. Length of the string does not exceeds 100.

Output

Print “Panic!” if the message is panic, or “No panic” otherwise.

Example

standard input	standard output
AAA!!!	Panic!
AAA	No panic
aaa!!!	No panic

Problem J. JokeCoin

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

The JokeCoin is the new cryptocurrency.

Vasya is planning to start working with this currency. People who are mining this currency receive the schedule of the data blocks for the day.For each data block Vasya knows time interval to process this data block, and the reward in JokeCoins for successful mining of this block (i.e. when block was processed at given interval without interruptions). Vasya can freely select the blocks to process; mining of the next selected block may start at the same second, when mining of current block is finished.

Vasya don't have enough resources to process more than one block simultaneously. Additionally, per each second used for mining, Vasya pays fixed amount of JokeCoins to the power company.

Vasya wants to know, if he will have any profit for work with JokerCoin, so he asks you to write the program which can calculate maximum expected profit for Vasya using given schedule.

Input

First line contains two integers N and C ($1 \leq N < 86\,400, 0 \leq C \leq 1000$) — number of blocks in the schedule and amount of JokeCoins Vasya pays to the power company per second of mining. Next N lines describe blocks.

Each of those lines contain starting time of mining and ending time of mining in the format `HH:MM:SS` between `00:00:00` and `23:59:59`, inclusively, difference between starting and ending time for one block is atleast 1 second and the bonus P ($0 \leq P \leq 10^5$) in JokeCoins for successful mining of this data block.

Output

Print one integer — maximal possible profit, or 0, if profit is zero or negative.

Example

standard input	standard output

`4 0`
`03:00:00 10:10:00 20`
`01:00:00 02:30:00 50`
`16:10:00 19:00:00 100`
`02:30:00 22:00:00 200`
	`250`

`3 1`
`16:59:00 17:00:00 100`
`01:01:01 01:01:11 20`
`12:00:00 13:00:00 3601`
	`51`

`4 10`
`00:00:05 00:01:55 1100`
`00:00:10 00:00:21 100`
`00:01:50 00:02:00 80`
`23:59:00 23:59:05 40`
	`0`

Problem K. King and ICPC

Input file:	*standard input*
Output file:	*stadnard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

In the kingdom are N ICPC teams, numbered sequentially between 1 and N, each containing of 3 contestants. Each contestant have KF (KingForces) rating — an integer.

King plans to welcome one contestant per team from teams with numbers between l_i and r_i inclusively to his palace in such a way that:

1. Sum of ratings of all contestants in the palace is divisible by an integer D (favorite integer of the king).
2. While previous holds, sum of ratings of all contestants shall be maximum possible.

The King still not decided, which interval to select, so he wants to know answer (maximal summary rating) for several intervals.

Input

First line of the input contains two integers N ($1 \le N \le 5 \cdot 10^4$) and D ($1 \le D \le 50$).

Each of the next N lines contains description of one team — ratings of the teammates. Rating is an integer between 0 and 10^9, inclusively.

Next line contains one integer M — number of the intervals ($1 \le M \le 3 \cdot 10^5$).

Each of the next M lines contain description of one interval — two integers l_i and r_i ($1 \le l_i \le r_i \le N$).

Output

For each interval print the maximum summary rating divisible by D. If it is impossible to choose contestants from teams in such way, print -1.

Example

standard input	stadnard output

2 2
0 1 3
1 2 3
3
1 1
2 2
1 2
	0
	2
	6

3 3
0 3 6
1 4 7
1 2 3
4
1 1
1 2
1 3
2 2
	6
	-1
	15
	-1

Problem L. Longest Simple Paths

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 mebibytes

Given a connected, undirected graph G, define an *shortest-path tree* rooted at vertex v is a spanning tree T of G, such that the path distance from root v to any other vertex u in T is the shortest path distance from v to u in G.

We will construct a shortest-path tree using the following method:

- We consider a shortest-path tree rooted at node 1.
- For every node i in the graph G, we choose a shortest path from root to i.
- If there are many shortest paths from root to i, we choose the one that the sequence of passing nodes' number is lexicographically minimum.
- All edges on the paths that we chose form a shortest-path tree.

Now we want to know how long are the longest simple paths which contain exactly k nodes in the shortest-path tree and how many these paths? Two simple paths are different if the sets of nodes they go through are different.

Input

The first line of the input contains three integers n, m, k ($1 \le n \le 3 \cdot 10^4$, $1 \le m \le 6 \cdot 10^4$, $2 \le k \le n$), denoting the number of nodes, the number of edges and the nodes of required paths respectively.

Then m lines follow, each of them contains three integers a, b, c ($1 \le a, b \le n$, $1 \le c \le 10^4$), denoting there is an edge of length c between a and b.

Output

Print two numbers — the maximal length of the path and the number of longest paths. You may assume that at least one such a path exists.

Example

standard input	standard output

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