Finals!

16th International 24-hour Programming Contest

http://ch24.org

DIAMOND GRADE SPONSORS

GOLD GRADE SPONSORS

ORGANIZER

Contest

Welcome to the 0x10th International 24-hour Programming Contest!

Rules

The contest starts at 2016-06-25 09:00 CEST and ends at 2016-06-26 09:00 CEST.

No solution can be submitted after the 24 hour time is up.

Web server

General contest related information will be available on our web server at http://server.ch24.org/.

Submission site

The same submission system will be used as during the Electronic Contest. It will be available at http://server.ch24.org/sub/.

Task summary

There are various kinds of problems, with various scoring rules and submission methods. Here we provide a short summary:

Task	Web submission	Interactive	Score decreases with time	Penalty for wrong answer	Time delay after fail/pass	Scaling	Max score
A (Maze runner)	Yes	No	Yes	-5	60/60	No	1000
B (Corporate Issues)	Yes	No	Yes	-5	60/60	No	1000
C (Crack it)	Yes	No	Yes	-20	60/60	No	1000
D (Flower shop)	Yes	No	Yes	-5	60/60	No	500
F (Hexasudoku)	Yes	No	Yes	-5	60/60	No	1000
H (High powers)	Yes	No	Yes	-5	60/60	No	1000
I (Robbery)	Yes	No	Yes	-5	60/60	No	1000
K (Tiles)	Yes	No	No	-5	60/60	Yes	1000
L (Brackets)	Yes	No	Yes	-5	60/60	No	1000

- Web submission: A static output file must be uploaded through the submission site.
- Interactive: During the solution or the submission, either network communication or other kind of interaction is necessary.
- Score decreases with time: Submitting at the end of the contest is worth 70% of what would be awarded at the beginning.
- Penalty for wrong answer: Wrong answer gets -5 points (different value may be specified explicitly in the task description).
- Time delay after fail/pass: Duration in minutes while no new submission is accepted after a wrong/correct answer.
- Scaling: The score for this problem may change over time depending on submissions by other teams. (Note that your last submission is considered and not your best one.)

Ports

Port	Task	Service description	Connection direction server <-> team
`80`	-	web	←
`6667`	-	irc	←
`u53`	-	dns server	←
`u67,u68`	-	dhcp server	←

Ports starting with `u` are UDP ports.

The UDP destination ports should be opened by the teams (if they are interested in the information).

All other services are hosted on `server.ch24.org`.

Contact

General contest related information and data will be published on the web at http://server.ch24.org/.

Important announcements will be made on the `#info` irc channel and will be published on the web as well.

For general discussions and questions join the `#challenge24` irc channel.

There will be separate channels for task related problems as well: `#A`, `#B`, etc.

Prologue

Eesteciana Jones wants to found the best beer factory in universe. He needs your help to achieve this goal!

A. Maze Escape

For this problem, you'd have to tap into the character of Eesteciana Jones (pun intended), a very well known adventurer. He managed to get himself stuck into a maze that contains something very valuable to him - diamonds. Just to make things a bit spookier, he has limited time to get out of the maze: there is lava right behind him that is definitely going to kill him if he isn't too careful. The maze is also complex, meaning there are walls at some points in it. Eesteciana has to avoid them as they would certainly mean his doom since he won't be able to run away from the lava in time if he reaches a wall. However, being as greedy as he is, he also wants to grab the maximum amount of diamonds before he gets out of the maze.

Also, let's talk a bit more about the maze. It itself is kind of self explanatory - since it contains diamonds its shape also resembles a diamond.

For example, a maze of size 3 would look like this:

```
          2;0
     1;0       3;0
0;0       2;1       4;0
     1;1       3;1
          2;2
```

A maze of size 5 on the other hand would look like this:

```
                    4;0
               3;0       5;0
          2;0       4;1       6;0
     1;0       3;1       5;1       7;0
0;0       2;1       4;2       6;1       8;0
     1;1       3;2       5;2       7;1
          2;2       4;3       6;2
               3;3       5;3
                    4;4
```

As you may observe, each column of the maze is enumerated with the numbers from 0 to 2 * (size - 1). The size itself represents the maximum length of a column, as we can see in the second case it's the 4th column and has exactly 5 blocks on which Eesteciana can step on. He always starts at 0;0 (that's the entry point) and the exit is at 8;0. He can also only move to adjacent blocks that make him escape the lava - for example if he's on block 1;0 he may move to block 2;0 or 2;1, but not to the block 1;1 or back to 0;0 since the lava would catch up to him. If there is a wall on a certain block he may not move to that particular block.

You will also be given an array of strings representing the positions the diamonds as well as an array of strings representing the positions of the walls (their indices will correlate to the ones represented on the diamond like map). So your task is to provide the maximum number of diamonds Eesteciana can get AND get out alive or just return "-1" if he'd die horribly anyway.

Note: The wall blocks will be given in an interval to avoid writing too many strings.

Example:

Let us assume that the size of the maze is 3 and that there is a diamond on 2;0. There is also a wall on 2;1.

The most optimal solution for this example would be to go to the blocks as follows:

```
0;0 -> 1;0 -> 2;0 -> 3;0 -> 4;0
```

This will both let you escape without hitting any walls, and getting the only diamond that the maze has (on 2;0). This means that the solution to this problem is going to be 1.

Input

The first line of input contains an integer M ($3 \le M \le 101$) which is guaranteed to be an odd number.

The next line will contain an integer D ($1 \le D \le 50$) representing the number of diamonds.

The following D lines will contain strings formatted as "x;y" and each string will represent the coordinates of the given diamond according to the maze notation we mentioned earlier. It is guaranteed that each of the strings is going to be unique.

The next line will contain an integer W ($1 \le W \le 50$) representing the number of walls in the maze.

The next W lines will contain strings representing each walls, formatted as "x;y;z". This means that there is a wall on the xth column on the indices from y to z (inclusive). It is guaranteed that no walla overlap.

Output

Output the most diamonds Eesteciana can get while still being able to survive the maze, or output -1 if there is no way for him to escape the lava and his imminent death.

Example input	Example output
11 9 14;3 8;1 11;9 13;5 6;5 13;7 13;1 5;4 10;3 3 11;6;6 2;1;1 6;1;2	4

B. Corporate Issues

General Phillip II's army or the phalanx contains r*c warriors organized in a matrix. The strength of the warriors is distributed in a very strange way. The strength of warrior at position (i,j) is defined as (a * p[i] + b * p[j]) % m where p[i] is the i-th prime number (1 is not a prime number!). The first warrior is at position (1,1).

He was invited to take part in a competition and he needs to send some group of warriors. Due to his great experience he knows that he needs to send warriors which have been close together in their battles.

A warrior's neighbours are those that are to the left, right, in front or behind him. A group of warriors must contain more than one warrior and there must exist a path between all warriors by only visiting neighbouring warriors which are also in the group. Because a group of warriors requires a lot of teamwork, its strength is defined as the minimum strength of all it's warriors.

Return the sum of maximal strength of a group for all sizes of groups (from 2 to r*c inclusive) modulo 10^9+7, meaning the sum of the best group of size 2, plus best group of size 3 etc...

Note: One warrior can be selected for multiple groups of different size.

For example, given a = 7, b = 5, m = 5, r = 3, c = 3 (where r and c are the dimensions of the matrix), the solution would be 11.

Input

The first line on input contains five integers, a, b, m, r and c

a - value of parameter a, used for calculating strengths of warriors

b - value of parameter b, used for calculating strengths of warriors

m - value of parameter m, used for calculating strengths of warriors

r - number of rows of the army

c - number of columns of the army

Output

Output the sum of the best group for sizes [2, r*c] modulo 10^9+7

Constraints:

a,b and m will be between 1 and 200000000 ($2*10^9$)

r and c will be between 2 and 900 inclusive

All indices are 1-based

Example input

```
11 7 7 5 9
```

Example output

```
102
```

C. Crack the ZIP

You are given a multilevel passworded zip file. Crack it!

Input

The zip file itself. You might need to unlock the previous level, the next input.

Output

Though you can check the password for each level locally, to earn points upload your passwords as soon as you get them.

Example

If the password for level **01** (i.e. for 01.zip) would be *SeCrEtPaSsWoRd01*, you should upload **01.out** with one line containing the string *SeCrEtPaSsWoRd01*.

D. The Virág’s flower shop

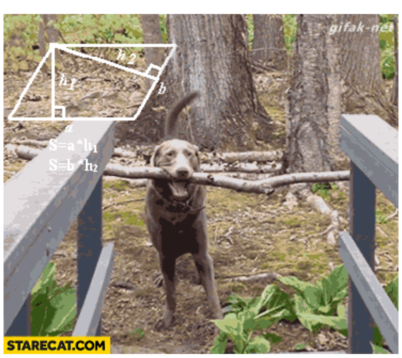
You are Bálint Virág, a grandson of great Zsigmond Virág, owner of the most famous Hungarian flower shop. Zsigmond created this business when he was very young and has been running it by himself ever since. He put his whole heart into making it special and unique - now it attracts both the locals and the tourists visiting the beautiful city of Budapest.
Zsigmond is planning to retire soon and he wants you to take over the family business from him. You have been helping him for a several weeks now, but you still feel like floristry is a great and mistical art, especially when you carefully observe your grandfather while doing his art. And you are sure that mastering it on the level of your grandfather will take you years of practice.

So today, you came early to the shop, like usually, to clean it up before the first customers will come. Surprisingly, you find a note on the counter, saying:

`"Bálint, life is too short and I saw so little of the world... I decided to go to the Florida for a month trip. #yolo Hope you can run the shop without me!!1".`

Hmm.
What is special about your grandfather is that, apart from being a very talented florist, he is also a big fan of riddles. One big riddle, that is definitely the main reason of popularity of Virág’s flower shop, is the **complicated system of selecting the flowers for bouquettes**. Of course, your grandfather didn’t share the exact recipe for creating these special bouquets with you (why would he...) but luckily you were a careful observer and figured out that there are some rules and patterns behind how he runs his business.

- The Virág’s flower shop is well known to sell **a finite number of bouquettes per day**. This number is known for your grandfather and it can be named as *b*.
- Every bouquet consists of **different types of flowers**. There cannot be e.g. two daisies in one bouquet.
- Every bouquet is **unique.**
- Every bouquet is **equal in quantity**, e.g. contains the same amount (*n*) of different type of flowers.
- There are plenty (*t*) different types of flowers.
- The flowers appear to be distributed throughout the bouquets randomly, but... here’s your grandfather’s greatest secret and where the magic happens: **there is always exactly one common flower type between every two bouquettes sold this very day!** How’s that possible? Only Zsigmond knows. (And luckily, you, before the first customers come.)



http://starecat.com/content/wp-content/uploads/dog-calculating-how-to-carry-a-stick-on-a-bridge.gif

Your task is to **generate the largest possible set of bouquettes that safisfies the above rules** with given n and t. How many bouquettes should be sold that day (b) - it's up to you to figure that out. Also, please keep in mind that the solutions with redundant flowers types (ones that are not really used) are not accepted.

Input

Files from 0.in to 9.in contain two numbers: first one is n - number of flowers per bouquet and the second one is t - different types of flowers in total.

Output

First line of output contains two numbers - first is b - the total number of bouquettes to be sold that day and the other one is n - number of flowers per bouquet (the same as in input). Second line is concatenated list of flowers and bouquettes. Different types of flowers should be represented as numbers (e.g. if we have three types of flowers they should be represented as 1, 2 and 3).
For example, if we have b=3 bouquettes with t=2 types of flowers per card using s=3 different flowers: [1,2], [2,3] and [3,1] we would write the following input:

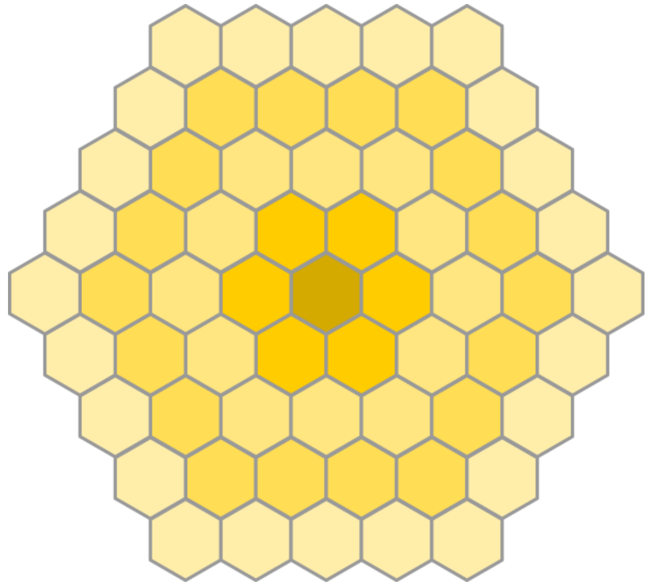
Example input

```
2 3
```

Example output

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

F. Hexasudoku



Rules of hexasudoku:

- The board is honeycomb-shaped, as seen on the picture on the right. There are 61 cells.
- Rows and diagonals have different lengths, meaning that there are rows and diagonals that are 5, 6, 7, 8 and 9 cells long.
- Each row, left diagonal and right diagonal has to contain **unique set of numbers.**
- Since the rows have varied length, **you don't know what numbers are in the row** (except from the longest one containing 9 digits). However, there are limited possibilities of numbers than can be present in cells:

```
+------------+------------------------------------------+
| Row length |          Possible sets of numbers        |
+------------+------------------------------------------+
|          9 | 123456789                                |
|          8 | 12345678 or 23456789                     |
|          7 | 1234567 or 2345678 or 3456789            |
|          6 | 123456 or 234567 or 345678 or 456789     |
|          5 | 12345 or 23456 or 34567 or 45678 or 56789 |
+------------+------------------------------------------+
```

In other words, the numbers present in shorter lines have to come from a set of consecutive numbers.

- Every puzzle has **only one** proper solution.

Your task is to **solve ten hexasudoku boards**.

Input

In order for input to be easier to process, the real "honeycomb" structure of hexasudoku (as seen on the left) has been simplified to one on the right - no left shifts are present:

```
    1 2 3 4 5            1 2 3 4 5
   6 5 4 3 2 1           6 5 4 3 2 1
  1 2 3 4 5 6 7          1 2 3 4 5 6 7
 8 7 6 5 4 3 2 1         8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9    >   1 2 3 4 5 6 7 8 9
 8 7 6 5 4 3 2 1         8 7 6 5 4 3 2 1
  1 2 3 4 5 6 7          1 2 3 4 5 6 7
   6 5 4 3 2 1           6 5 4 3 2 1
    1 2 3 4 5            1 2 3 4 5
```

The inputs from 0.in to 9.in are in this form. Zeroes mark the empty places.

Output

Your output should be formatted exactly the same as the input, but the zeroes should be filled with proper values. So your final solution (for each hexasudoku board) should look something like that:

```
1 2 3 4 5
6 5 4 3 2 1
1 2 3 4 5 6 7
8 7 6 5 4 3 2 1
1 2 3 4 5 6 7 8 9
8 7 6 5 4 3 2 1
1 2 3 4 5 6 7
6 5 4 3 2 1
1 2 3 4 5
```

Example input

```
0 0 2 0 0
3 1 0 0 2 4
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 6 8 0 0 0 7 3 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0
```

Example output

```
5 3 2 4 1
3 1 5 6 2 4
2 4 6 7 8 5 3
1 5 7 8 4 3 6 2
4 6 8 2 9 1 7 3 5
2 4 3 6 7 8 5 1
3 5 7 8 2 6 4
1 3 6 5 4 2
5 2 4 1 3
```

H. High Powers

Given numbers N and M, find the biggest power of M so that N! is divisible by it.

N! represents the factorial notation. Also, N! = N * (N - 1) * (N - 2) * ... 2 * 1.

Example:

Let N = 10 and M = 6.

We know that 10! in fact equals 3628800.

For the sake of explanation, let us write all of the powers of 6 that divite 10!:

6^0 = 1, which divides 3628800.

6^1 = 6, which divides 3628800.

6^2 = 36, which divides 3628800.

6^3 = 216, which divides 3628800.

6^4 = 1296, which divides 3628800.

6^5 = 7776, which does not divide 3628800.

This means that the highers power of 6 that divides 10! is 4. Thus, the answer is 4.

Input

The first line of input contains one string (1 ≤ N.length() ≤ 15) - the number N for which we would need to find N! (this in essence means that as a number, $1 \le N \le 10^{15}$).

The second line of input contains an integer (1 ≤ M ≤ 100), representing the number M.

Output

Output the biggest power K of M so that M^K still divides N!.

Example input	Example output
74 2	71

I. Robbery

So, you're a robber that enters a bank through its roof. Fortunately (for the owners of the bank at least), you aren't really a good robber so you did not remember to map out the bank beforehand, and thus you know nothing about its construction (or in which room you land in). The bank on the other hand is constructed of a lot of rooms with very big walls in between them - walls you can't possibly go through or break or anything. In each room there are coins (which just so happens you're after and would do anything to get as many of) which are spread out on the ground of the room. Now, since the police alarm goes off the moment you enter, you only have enough time to search through one room and gather the coins in it before you make your grand escape. This means that you won't be able to go to other rooms during your robbery. If given the way the bank is constructed, the positions of the coins in it as well as your starting position (where you enter) you need to find the number of coins you can get before making a run for it. Additionally, you will be given multiple queries in each test case for all of which you need to provide the answer.

Details and assumptions:

The bank will be represented to you as a matrix consisting only of the characters ' . ' and ' * '. ' * ' means that there is a wall in that particular position whereas ' . ' means that there is a free spot (one you can step on and move to).

Suppose you are currently at a certain spot labeled (x,y). You are allowed to move to spot (x1,y1) if and only if (x,y) and (x1,y1) are adjacent and both have the character ' . ' on them. This will become clearer as we explain the test cases.

Input

The first line of input contains four integers, n, m, s and c.

n - an integer representing the number of rows of the bank
m - an integer representing the number of columns of the bank
s - an integer representing the number of starting positions (queries) you will be given
c - an integer representing the number of coins you will be given

The following n lines will contain strings that consist of m characters each. The jth character of the ith line means that the point of the bank labeled as (i,j) has that character.

The following c lines will contain a pair of numbers each. The pair x,y means that there is a coin on the point (x,y) in the bank.

The following s lines will contain a pair of numbers each as well. The pair x,y means that the starting point of the given query is the point (x,y) in the bank.

Output

Your output should be s lines. The ith line should be a solution to the ith query given above.

Constraints:

$1 \le n,m \le 1000$
$1 \le s \le \min(n*m,100000)$
$0 \le c \le$ the number of ' . ' characters in the bank illustration
Each one of the starting positions is guaranteed to be on a point that is not a wall
Each one of the coin positions is guaranteed to be on a point that is not a wall
All indices will be 1-based

Explained example:

Input:

5 5 10 4

```
*****
*...*
*..**
*.*.*
*****

4 2
2 2
2 3
3 3
2 4
4 2
2 2
2 3
2 2
2 2
2 4
3 2
2 2
2 2


The output here should be:

4
4
4
4
4
4
4
```

```
4
4
4
4
```

Let's put a ' C ' in every point where there's a coin. The bank matrix would look something like this:

```
*  *   *   *  *
*  C  C  .  *
*   .  C  *  *
*  C  *   .  *
*  *   *   *  *
```

So let's just illustrate the first query. The point where we enter is (2,4). Now from (2,4) we can get only to (2,3), from where we can travel to (3,3) or (2,2) and so on. In short, from (2,4) we can get to the points (2,3), (2,2), (3,2), (3,3) and (4,2). However, this is not the only room in the bank. The point (4,4) is a room on its own due to the fact that there is no way to travel to (4.4) from (2,4). Fortunately, all of our queries are in the room with the coins so that is why all of the answers are 4 (we can get from any point of that room to any other point in that room). If we had (4,4) as a query the answer to that one would be 0, as you can't get to the room with the coins in any way.

Example input

```
5 6 3 3
******
*..*.*
******
*....*
******
2 3
2 5
4 5
2 2
2 5
4 3
```

Example output

```
1
1
1
```

K. Tiles

We've reached the next level in tile switching challenges. A team of trained programmers and chemical experts arrived at a storage facility to prevent a serious accident. Inside the facility there are different kinds of explosive materials stored in cubic boxes. These chemical have strange properties: each side of each box has a number(between 0 to 3) representing different reaction types. If two boxes are next to each other, and the neighbouring sides have the same properties (represented by the same number), then a chain reaction might go off and the world would end. We would like to prevent such things, so programmers must come up with an algorithm and idea to save everybody. During the process they have to switch cubes to achieve a space where no neighbouring side has the same colour. Note that time is running up, so you have to minimize the number of steps!! The objective is that no 3D cube should have sides that have similar colour as the next cube' side touching the examined side. This applies for all 6 sides of every cube.

$$$task

Input

Output

Example input

```
Example input
2 3 3
2 1 3
0 0 2
1 3 3
3 2 1
0 1 3
2 0 1
```

Explanation: first row means the upper bound (maximum coordinates in each direction): X = 2, Y = 3, Z = 3 we have two layers (x=2) each with size 3x3 others rows represent elements (for example): there's a cube at coordinates (1, 1, 1), its value is 2: meaning that this 1x1x1 unit has chemical type 2 there's a cube at coordinates (1, 1, 2), it's value is 1 there's a cube at coordinates (1, 2, 1), it's value is 0 there's a cube at coordinates (2, 1, 1), it's value is 3 etc.

Example output

```
Example output(two x,y,z values, you change the first cube with the second):
1 3 3 1 2 1
```

Explanation: The cube located at (1, 3, 3) will be switched with the one at (1, 2, 1) therefore every neighbouring side has different chemicals at the end.

L. Brackets

You are given a string S of '(' and ')' characters of length n. You are to find the number of its longest regular bracket subsequences. Answer should be output modulo 104857601.

A regular bracket sequence is a bracket sequence that can be transformed into a correct arithmetic expression by inserting characters '1' and '+'.

For example, the longest regular bracket subsequence of the string S="())())(" has length 4. There are four such subsequences: [0,1,3,4], [0,2,3,4], [0,1,3,5], [0,2,3,5]. So, the answer for this case is 4.

Input

The first line of input contains one integer ($1 \leq T \leq 10^6$) - the number of testcases. Each of the next T lines contains a string consisting of bracket characters. The total length of all strings doesn't exceed $2*10^7$.

Output

Output T lines with answers for testcases, one integer per line.

Example input	Example output
2 (()) (()))((())	1 9