Problem A. GOV-internship

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

Hello Denis,

Your internship in Team.GOV is over. Unfortunately, we cannot make you a full-time offer. Don't be upset, because you are neither the first nor the last one. It was a great pleasure to work with you. We wish you best of luck.

Regards, Vadim Kantorov, the captain of Team.GOV

A freezing November night in the year of 2010. The dormitory of Department of Mathematics and Mechanics of Ural State University. Dim monitor light. That's how Den Mukhametianov (the 10th team member in succession) left Team.GOV.

At the beginning of September Den returned from hot summer vacations. He was calmly browsing the photos from a trip when he got an email "You are invited to an interview with Team.GOV". A year before Den could only dream of this.

The picture emerges in the mind as if it all had been yesterday. It was a small cozy room, Vadik settled comfortably in a rocking chair. Den coughed awkwardly and shitfed his feet.

Vadik looked up and smiled. "What experience have you got?"
"Half a year in Ural SU AirBug and half a year in Ural SU Quickov."
"Hm, not bad. What's your favorite topic?"
"String algorithms."
"Well, then straight to the point then," Vadik interrupted Den. "Let's pick a problem. Do you know what is Hamming distance between two strings of equal lengths?"
"Yes, it's the number of positions on which the characters of the strings are not equal."
"Exactly! Let's define the distance from a pattern p to a string s as the sum of Hamming distances from p to all substrings of s of length $|p|$. In the string and in the pattern some characters are erased. You need to recover the erased characters in such a way that the distance from p to s is minimal."

Input

The first line contains the string s, the second contains the pattern p. Both strings are not empty and their lengths don't exceed 1 000. Strings are comprised of the following characters: '`0`', '`1`' and '`?`'. Here, '`?`' stands for erased characters that you need to recover. Length of p doesn't exceed length of s.

Output

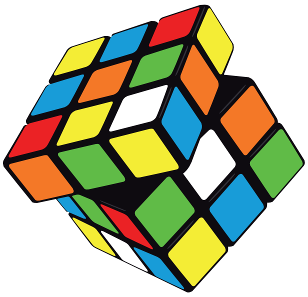
The first line should contain the minimal distance from p to s after all erased characters are recovered. In the second and in the third line output s and p respectively where each character '`?`' is replaced with either '`0`' or '`1`'.

Example

input.txt	output.txt
00? 1?	2 000 10

Problem B. Cube in a Cube in a Cube

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes



Denis had a Rubik's cube covered with pretty patterns on his table. When Denis quit the room, Vadik took the cube and made a few rotations. But he got scared suddenly: what if Denis get angry about the broken patterns? Vadik wanted to restore the cube to the original state, but could not find a way.

Vadik heard somewhere that if one repeats a sequence of rotations again and again, one would get the original pattern at some point. Given this is true, it is possible to revert the cube until Denis returns. How many times does Vadik need to apply his sequence of rotations in order to revert the cube to the original state?

A Rubik's cube consists of $3 \times 3 \times 3$ smaller cubes. The faces of the smaller cubes which lie on the faces of Rubik's cube are painted in different colors; for the purpose of this problem, all painted faces of all smaller cubes are considered pairwise different. A rotation of some face of the cube affects 3×3 smaller cubes which lie on that face (see picture). The smaller cube at the center of Rubik's cube remains stationary. We consider two states of the cube to be the same if all painted faces of smaller cubes have the same colors.

Input

The description of rotations applied by Vadik in the form of a string consisting of letters '`U`', '`D`', '`L`', '`R`', '`F`' and '`B`' (upper or lower case). They mean rotations of top, bottom, left, right, front and back faces, respectively. Lowercase letters mean 90-degree clockwise rotations, uppercase letters mean 90-degree counterclockwise rotations. The clockwise and counterclockwise directions of rotating a face are measured as if one looked at that face. The string is not empty and its length does not exceed 10^5.

Output

Output the number of times Vadik needs to apply his sequence of rotations in order to get the original pattern.

Examples

input.txt	output.txt
`Ud`	`3`
`RRRR`	`0`

Problem C. Laser Technologies

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

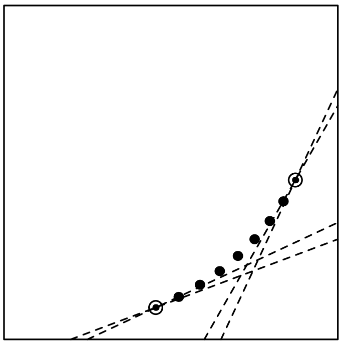
Almost every gift shop sells cubes and balls of glass with beautiful images inside them. Young girl Anya wanted to learn the way these images are created, so she entered the department of laser technologies. Now she knows that if two laser beams of frequency 3.1415 PHz intersect inside the glass object, the glass will become opaque at the intersection point, although this opacity will not block other laser beams passing through this point.

Now Anya herself wants to draw an image inside a polygonal prism. She decided to start with a flat image located in a section parallel to the base of the prism. Unfortunately, Anya spilled her tea on the second laser emitter, and it broke down. However, Anya knows that if two laser beams will pass through some point consecutively within a short period of time, this point will also become opaque. Now she wants to wants to move the laser emitter all the way along the perimeter of the section. Laser beam will always lie inside the plane of this section. Laser beam should always be directed to a special receiver. Anya decided to move the receiver in the same way along the perimeter of the section, with the same speed and in the same direction as the emitter.

The frequency of the laser beams is extremely high, so the opaque points will be very close to each other, and will be recognized as parts of continuous curves. Anya wants to know the total length of all these curves, and asks for your help.

More formally, the section of the prism is a convex polygon with n vertices. Initially, Anya places an emitter at some point of its perimeter. She then places the receiver at some other point of the perimeter such that the distance between the emitter and the receiver measured along the perimeter is d. After that, Anya starts to move both the emitter and the receiver with equal constant speed in one direction along the perimeter. She stops when the emitter and the receiver move to their initial positions.

The formal definition of the resulting opaque parts of curves is as follows. Consider some moment t when the emitter is at point A and receiver is at point B. In a near moment $t+\varepsilon$, the emitter (moving along the perimeter) arrives at some point A' while the receiver (moving with the same speed along some other part of the perimeter) arrives at some point B'. In general case, there will be exactly one intersection point of segments AB and $A'B'$. Now let $\varepsilon \to 0$ to get the point which becomes opaque at the moment t. Do that for all possible moments t to get all the points which become opaque. This set of points in fact can be viewed as a set of curves. Your task is to find their total length.



On the illustration, the perimeter is a square, dashed lines represent the laser beam at different moments of time, and bold points are the points which become opaque as the emitter and the receiver move along the perimeter. The distance between the emitter and the receiver (measured along the perimeter) remains equal to the constant d. Note that it does **not** mean the lengths of dashed lines are equal. As you can see, the opaque points lie on a certain curve.

Input

The first line contains an integer n ($3 \leqslant n \leqslant 10\,000$). Each of the following n lines contains two integers which are the coordinates of vertices of the polygon in counter-clockwise order. The coordinates don't exceed 10^6 by their absolute values. It is guaranteed that no three vertices lie on the same line.

The last line contains a positive integer d. It is guaranteed that d doesn't exceed half of the length of polygon's perimeter and is strictly greater than the longest of its sides.

Output

Output the total length of all curves formed by the opaque points with absolute or relative error not exceeding 10^{-6}.

Examples

input.txt	output.txt

3
-1 0
1 0
0 10
11
	22.3730148246

4
0 0
10 0
10 10
0 10
11
	56.1255583559

Problem D. Negotiations with Parthians

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

Ancient Greeks and Romans thought that gods of the underworld like even numbers, and the gods who protect the ones who were alive like odd ones. In 123 B.C., Rome was threatened by the war with Parthian Empire. The emperor Adrian did not want the bloodshed; therefore he went to carry the negotiation with parthians. In order get the gift of great eloquence from the gods, he wanted to sacrifice a herd of n sheep. Of course, n was an odd number. Thinking about that, Adrian realised that he should give the equal number of sheep as a sacrifice to each of the gods. Otherwise, a god who gets less sheep than some other god becomes furious at Adrian, and this might prevent him to solve the problem in the peaceful way.

The emperor called for Theon of Smyrna, the mathematician, and ordered him to divide sheep equally. Theon of Smyrna was also superstitious, that's why he thought that the amount of gods not only should be odd, but also should have odd amount of positive divisors. Apparently, he was able to solve the problem and negotiations with Parthia finished successfully. Can you solve this problem?

Input

The only line contains an integer odd number n that is a total amount of sheep to sacrifice ($1 \leqslant n \leqslant 10^{18} - 1$).

Output

Output the number of gods to receive a sacrifice. If this problem has several answers, output the maximal one.

Example

input.txt	output.txt
45	9

Problem E. Trade Guilds of Erathia

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

The continent of Antagarich was colonized slowly. Long ago its northern part was inhabited by the elves of Avlee. Later, the hot southern desert of Bracada was occupied by the white mages. At the same time, necromancers settled in Deyja, a land to the north of Bracada and to the south-west of Avlee. Although white and dark mages didn't really like each other, each group had some artifacts that the other group would be happy to buy. As a result, the trading relationship between Bracada and Deyja grew stronger, and soon the mages built a very busy trade route between these lands.

Erathia was founded later, and at first it was stretched along this route. At that time Erathia's economy was based solely on trading, so new trading guilds appeared all the time. Each of the guilds was present in a few cities which were consecutively situated along the route. Caravans of each guild travelled between all pairs of cities of that guild equally often.

The state's treasury was replenished by fees collected from all the caravans moving along the trade route. There was a fee for each route segment connecting two neighboring cities, and this fee could change over time. For example, the fee could be decreased in the areas of frequent goblin attacks, or increased in the areas with high traffic.

Loins, the royal treasurer, studies Erathia's economy and tries to predict the profit of trade guilds. He wants to know the amount of money paid in fees by each guild. He has a chronologically ordered list of documents that contains all the royal orders changing the fee and all the papers establishing new guilds. This data should be used to calculate the average fee paid by a caravan of a given trade guild.

Input

The first line contains the number n of cities in Erathia and the number m of documents collected by Loins ($2 \leqslant n \leqslant 10^5$; $1 \leqslant m \leqslant 10^5$). The following m lines describe the documents of two possible types:

- "`change` a b d": the fee for travelling along each route segment between cities a and b changed by d gold coins (if d is positive, the fee increased; if d is negative, the fee decreased);
- "`establish` a b": a new guild which is present in all cities between a and b was established.

All numbers are integers; $1 \leqslant a < b \leqslant n$; $-10\,000 \leqslant d \leqslant 10\,000$. Cities are numbered in the order they are located along the route: from Bracada to Deyja. The fee for travelling along a segment was never larger than 10 000 gold coins, otherwise merchants would protest. Of course, the fee was always non-negative. Before the first royal order changing the fee, it is equal to zero for all route segments.

Output

After each document establishing the new guild, output in a single line the average amount of fee paid by a caravan of this guild. The absolute or relative error should not exceed 10^{-6}.

Example

input.txt	output.txt

`4 5`
`change 1 4 2`
`change 1 2 -1`
`establish 1 2`
`establish 2 4`
`establish 1 4`
	`1.00000000`
	`2.66666667`
	`2.83333333`

Problem F. War and Peace

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

During the war in Erathia, an army of liches had to solve a difficult task of relocating its troops. Liches can teleport from one city to another quickly enough, but giving directions to thousands of liches in dozens of strongholds seems to be a real problem. This problem was especially difficult when the Great lich Sandro was powerful, as the army of liches grew huge.

Sandro noticed that only two of all the spells, *Curse of the Netherworld* and *Implosion*, are powerful enough. When somebody casts them, it can be felt everywhere in the country. He decided to use these spells as signals to his troops. When liches in a certain stronghold feel a certain spell, they should teleport to the predefined destination stronghold (or stay in their current stronghold). After a spell is cast, the liches in all the strongholds simultaneously teleport according to this scheme. Liches have no troubles with discipline, so this scheme works perfectly since it was introduced.

When Sandro died, the peace between people and the undead was proclaimed. The remaining liches were evenly distributed among all the strongholds in Deyja in case of a sudden attack, while the top undead commanders started to fight for power. Soon necromancer Vidomina and vampire Korbu became the most powerful among the undead. Their influence was almost equal, so none of them could become the only leader, and the country was split between them. Recently, Vidomina decided to start another war against Erathia and began to gather troops. As an experienced warlord, she understands the strength of Erathia's army, and now wants to gather all the troops in one stronghold prior to the attack. Meanwhile, Korbu thinks that this war can lead to total annihilation of the undead and wants to keep truce. He can't convince Vidomina, so he decided to prevent her from gathering all the troops together.

Casting signal spells requires a lot of energy, so they can only be cast once a day. Each noon Vidomina casts one of the spells, in order to gather the liches together. On the other hand, each midnight Korbu also casts one of the spells, trying to stop Vidomina. There are no other mages capable of casting these spells in Deyja, so no one else can move the troops. Vidomina casts her spell first. Assuming that both mages act optimally, find out whether Vidomina succeeds in gathering all the liches together.

Input

The first line contains an integer n, the number of strongholds in Deyja ($2 \leqslant n \leqslant 1000$). The i-th of the following n lines contains two integers a_i and b_i which are the numbers of strongholds where troops move from stronghold i after *Curse of the Netherworld* and *Implosion* spells, respectively ($1 \leqslant a_i, b_i \leqslant n$).

Output

Output "`War`" if Vidomina will succeed in gathering all the troops together, or "`Peace`" if Korbu will prevent her from doing that in any finite number of days.

Examples

input.txt	output.txt

`2`
`2 1`
`1 2`
	`Peace`

`2`
`2 2`
`2 1`
	`War`

Problem G. Alice and Bandersnatch

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

> *The Queen turned crimson with fury, and <...>*
> *screamed "Off with her head! Off—"*

After talking to the Queen of Hearts, Alice understood one important thing: one inappropriate word can make you beheaded. So, Alice started to cipher her dialogues with her new friend Bandersnatch.

Recently she happened to invent a cipher which is easy in use, yet secure. Alice uses a sequence of integers $x_1, \ldots, x_n$ ($0 \leqslant x_i \leqslant a-1$) as a key; here, a is the size of the alphabet. Instead of sending Bandersnatch a message $s = s_1 s_2 \ldots s_r$, she sends another message $t = t_1 t_2 \ldots t_r$. Here, the alphabetic number of letter t_1 is the alphabetic number of letter s_1 plus the integer x_1, the number of t_2 is the number of s_2 plus the integer x_2, and so on; naturally, the next after the last letter of the alphabet is the first one. After pronouncing the first n letters of her message, Alice uses her sequence from the beginning by using x_1, then x_2, and so on (for example, the number of t_{n+1} is the number of s_{n+1} plus x_1).

Alice wants to know the number of different keys for her cipher. The keys that can be obtained from each other by a cyclic shift (e. g., $[0, 1, 2, 0]$ and $[2, 0, 0, 1]$) are considered the same. Moreover, Alice is afraid of eavesdropping, so she doesn't want to use sequences with period less than n (these are the sequences of length n which can be represented as shorter sequences repeated some integer number of times, like $[0, 1, 0, 1]$).

Input

The only line of input contains integers a and n ($1 \leqslant a \leqslant 26$; $1 \leqslant n \leqslant 10\,000$).

Output

Output the number of different keys for Alice's cipher.

Examples

input.txt	output.txt
3 4	18
10 2	45

Problem H. Magic Cube

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

Kreegan's attack in 1165 (Xeen's chronology) was one of the hardest challenges in the history of Enroth. However, the artifacts found after this war lead to the intense development of magic in Enroth. The way some of these artifacts work is still unknown, and the best mages conduct serious research and experiments trying to perceive their magic power.

One of these artifacts is a magic cube which, according to veterans of that war, was used by kreegans to prolong the spells' action, thus saving the power of mages. This cube consists of many erudin crystals which are small cubes of equal size. Some faces of these crystals are covered with runes. Adjacent faces of two neighboring crystals contain the same rune on them. Faces that don't contain a rune on them form the faces of the whole assembled cube.

Unfortunately, now this artifact is disassembled and is represented by the set of crystals. Previous attempts to connect them were unsuccessful as the cube constantly broke into pieces during the process. However, the scientists discovered that the relative positions of certain pairs of crystals in the assembled cube play a great role in its magic power. Now the researchers want to analyze the runes on the faces of crystals to restore the way the cube should be assembled, and to calculate the distance between some pairs of crystals in the assembled cube. Crystals can be rotated and moved arbitrarily.

Input

The first line contains integers l and n which are the length of edge of a single crystal and the ratio of artifact edge's length to crystal edge's length ($1 \leqslant l \leqslant 100$; $2 \leqslant n \leqslant 50$).

The following n^3 lines describe crystals. Each line contains six numbers in range from 0 to 10^9 which denote the numbers of runes on the faces of a crystal according to the list of kreegan's runes by Patvin Darkenmore. Number 0 means that the face doesn't have a rune on it. Faces are described in the following order: bottom, top, right, left, front, back. It is guaranteed that any positive rune number is present exactly twice in the description of all crystals.

The next line contains an integer m which is the number of pairs of crystals researchers want to investigate ($1 \leqslant m \leqslant 10\,000$). Each of the following m lines contains integers a and b, which are the numbers of crystals you should find a distance between ($1 \leqslant a < b \leqslant n^3$). Crystals are numbered in the order they are described in input. It is guaranteed that there exists a way to assemble the artifact.

Output

For each pair of crystals, output on a single line a distance between the centers of these crystals in the assembled artifact. Absolute or relative error should not exceed 10^{-6}.

Example

input.txt	output.txt

1 2
0 8 73 0 0 16
0 9 0 73 0 1000
0 146 4 0 16 0
0 15 0 4 1000 0
146 0 1 0 6 0
15 0 0 1 2 0
8 0 17 0 0 6
9 0 0 17 0 2
2
1 5
2 8
	1.4142135
	1

Problem I. Last Season of Team.GOV

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

Four regional contests, twelve different teammates… Twenty sixth place is not that bad indeed, but World Finals would have been much better. And now the career is over, Team.GOV project is closed. The road back home to Yekaterinburg, graduation, and master's degree in France are ahead…

These thoughts were running around the head of Vadim Kantorov when the dean's voice drew him back to reality:
"Vadim, whom do you want to share a compartment with?"

A compartment on a sleeping car contains four main couchettes and two side couchettes. There are exactly $6n$ people in the Ural SU delegation, that's why the dean bought tickets for n consecutive compartments of a sleeping car. Every delegation member said what type of couchette they like more: main or side. Apart from that, everyone wants to share a compartment with their friends. The delegation needs to decide who occupies which compartment so that all these requirements are satisfied.

Input

The first line contains an integer n ($1 \leqslant n \leqslant 1000$). The second line contains a bit string of length of $6n$. i-th bit in this string is equal to 1 if the i-th delegation member wants to occupy a couchette of main type, and 0 if they want a couchette of side type. The next line contains an integer m that is the number of pairs of friends ($0 \leqslant m \leqslant 6n$). The next m lines contain all those pairs as integers a_j and b_j ($1 \leqslant a_j < b_j \leqslant 6n$). All these pairs are distinct.

Output

Output n lines. Each line should contain a space-separated list of delegation members who should occupy a compartment together. Each pair of friends should occupy the same compartment. The order of people in a single list can be arbitrary. The order of lists also does not matter. If there are multiple solutions, you can output any of them. It is guaranteed that at least one solution exists.

Example

input.txt	output.txt
2	1 2 3 4 5 6
001111001111	7 8 9 10 11 12
2	
1 2	
7 8	

Problem J. Fiborial

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

Consider a sequence F_i that satisfies the following conditions:

$$\begin{aligned} F_0 &= 1 \\ F_1 &= 1 \\ F_i &= i \cdot F_{i-1} \cdot F_{i-2} \end{aligned}$$

Find the number of different divisors of F_n. Since the answer can be large, output it modulo $10^9 + 7$.

Input

Input file contains the only integer number n ($1 \leqslant n \leqslant 10^6$).

Output

Output a single integer — the number of divisors of F_n modulo $10^9 + 7$.

Example

input.txt	output.txt
3	4

Problem K. Graveyard in Deyja

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	2 seconds
Memory limit:	64 mebibytes

Lich Sandro was alone at the ancient graveyard. The rulers of Deyja, and great necromancers, and liches, who died during the wars with elves, were buried there. Both ages and rigorous northern climate were ruthless to gravestones and monuments, some of them were broken in pieces, others were covered with fractures.

One of the graves amazed Sandro. It was astonishingly big and was situated at the center of the graveyard, it looked like someone very important in the history of Deyja was buried here. Maybe it was the first king of liches himself. Sandro wants to check if the writing on the gravestone contains the name of the first king. But the writing is difficult to read, only some letters could be recognized. Help Sandro use the recognizable letters to decide if his hypothesis can be true.

You should find the maximal number of times the name of the first king could possibly be encountered in the writing on the gravestone when all the letters were recognizable. More formally, the writing consists of letters and places where a letter could not be recognized. Insert the letters in these places so that the total number of occurrences of the name of the first king is maximized. The occurrences may overlap. String comparison must be case-sensitive.

Input

The first line contains a string of length n consisting of English letters and question marks. It represents the writing on the gravestone (question marks denote letters that can't be recognized). The second line contains a string of length m consisting of English letters. It represents the name of the first king of liches. It is guaranteed that $1 \leqslant m \leqslant n \leqslant 1000$.

Output

Output one integer: the maximal number of times the name of the first king could possibly be encountered in the writing when all the letters were recognizable.

Examples

input.txt	output.txt
`R?la?dDefe?tedHi?Br?t?erArc?i??ld` `Archibald`	`1`
`Arch?????rchibaldIronfist` `Archibald`	`1`