



Problem A. Abnormal Coins

Input file: stdin
Output file: stdout
Time limit: 1 second
Memory limit: 256 megabytes

If you have participated or solved any of the four previous AUT local contests, you may have noticed that first problem of each contest were all about abnormal new students of Amirkabir UT. This time we want to talk about coins in our problem-set but to respect traditions, let's start with **abnormal type of coins**.

Everybody knows what coins are, “A flat, typically round piece of metal with an official stamp, used as money.”. But as wikipedia “Not all coins are round. The Australian 50 cent coin, for example, has twelve flat sides. Some coins have wavy edges, e.g. the \$2 and 20-cent coins of Hong Kong and the 10 cent coins of Bahamas. Some are square-shaped, such as the 15 cent coin of the Bahamas. During the 1970s, Swazi coins were minted in several shapes, including squares, polygons, and wavy edged circles with 8 and 12 waves.”

Steve is a great coin collector but he has no polygon shaped coin in his collection. Recently he found another coin lover, Johnny, Who has a collection of polygon coins and fortunately agreed to deal with Steve. The deal is simple, e coins for each poly-coin with e edges. Steve can offer n coin(s) and wants to earn as many poly-coin types as possible to extend his collection. Two poly-coins are different if they differ in number of edges.

Input

Input contains a single integer n . ($1 \leq n \leq 10^8$)

Output

Output a single integer indicating the maximum number of distinct poly-coins Steve can earn.

Examples

stdin	stdout
2	0
3	1
100000000	14139



Problem B. Fake Coins

Input file: `stdin`
 Output file: `stdout`
 Time limit: 1 second
 Memory limit: 256 megabytes

Cristiano, as a leader of his team, works in the Security Lab of Central Bank of Real Mars (SLCBRM). They want to improve security of coins that Central Bank makes. They want to use string codes that will be carved on coins to determine whether the coin is fake or original. Cris's team suggested him an amazing idea for managing security of coins.

They have a base string called B and a sequence of numbers called S . They select some characters from the base string to generate the security codes. If the i -th element of sequence is S_i Then the i -th element of generated code is the S_i -th character of B i.e. ($B[S_i]$). As you can clearly see the length of the generated security string code is equal to the number of elements in the sequence.

Their sequences are very similar to fibonacci numbers but their first and second elements are not always 1! They select two different numbers as first and second element of sequence (first number is always less than the second number). Then the third number of sequence is the sum of first and second numbers, the forth number is the sum of third and second number and so on. This process stops when a new element is greater than the length of the sequence. All elements of the sequence must be less than or equal to length of the base string.

Cris read the algorithm of generating string codes and realized that it can be some problems with same string codes. He wants to know how many different string codes can be generated using his team's algorithm. Because Criastiano is busy with scoring for Real Mars Football team he asked you to solve his problem!

Input

First and only line of input contains the base string B ($3 \leq \text{len}(B) \leq 1000$). Base string is composed of digits, lowercase and uppercase Latin letter.

Output

Print one line containing the number of different security string codes that can be generated using Criastiano's team algorithm.

Examples

stdin	stdout
abba	5
abBA	6
FAKE0123456789original	217

Note

In the first sample the codes are:

- 1 2 3 -> "abb"
- 1 3 4 -> "aba"
- 1 4 -> "aa"
- 2 3 -> "bb"
- 3 4 -> "ba"



Problem C. Coin Graph

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 256 megabytes

First of all you must know that the name of this problem is irrelevant to what we want!

You are given an integer s . Find a simple connected graph such that the sum of lengths of the shortest paths between all pairs of its vertices is s .

Input

Input contains a single integer s . ($0 \leq s \leq 10^5$)

Output

Print "Impossible" (without the quotes) if there is no graph, satisfying the described conditions.

Otherwise in the first line print two space-separated integers n and m — the number of vertices and edges in your graph, respectively. Each of following m lines contain two space-separated integers u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$) — two vertices, connected by an edge in the resulting graph.

If there are several answers you can print any of them.

Examples

stdin	stdout
1	2 1 1 2
2	Impossible
3	3 3 1 2 2 3 3 1
48	7 6 1 2 1 3 2 4 2 5 3 6 3 7

Note

A graph is simple if it has no self-loops or multiple edges between the same vertices. All edges in a simple graph are unweighted and undirected.

A graph is connected if there is a path connecting every pair of vertices.



Problem D. Coin Table

Input file: `stdin`
Output file: `stdout`
Time limit: 3 seconds
Memory limit: 256 megabytes

Eli has got a square board as his birthday present. Some cells of this board are filled with coins and other cells are empty. Rashof (her Russian bear doll) is capable of collecting coins and can go right or down. She will put Rashof in the top-left cell of the table. If a rectangle of the table is blocked, which means Rashof can not pass or land on that rectangle at all, how many ways can Rashof take maximum number of coins and reach lower-right cell of the board?

Input

Each test case contains one positive integer n , the size of the board. Each of the next n lines contain n characters, '.' or 'C'. '.' is an empty cell and 'C' is a coin.

The next line contains Q , the number of queries. Each of the next Q lines contains four integers, $r1$, $c1$, $r2$ and $c2$, which are the top-left and bottom-right cells of the blocked rectangle.

For each query print the number of ways Rashof can take maximum number of coins.

$$1 \leq n \leq 1000$$

$$0 \leq r1, c1, r2, c2 \leq n - 1$$

$$1 \leq Q \leq 1000$$

Output

For each query print two space separated integers, the maximum number of coins Rashof can take without passing the blocked rectangle, and the number of ways it can do that modulo 1000000007.

Examples

stdin	stdout
6	9 7
C...CC	7 1
CC..C.	0 0
..C...	
CCC.C.	
CCC.CC	
CC...C	
3	
2 2 2 2	
1 0 4 2	
4 4 5 5	



Problem E. Volleyball

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

The FIVB Volleyball World Championship is an international men's and women's indoor volleyball competition. It is the oldest and most important of all the international events organised by the FIVB and must not be confused with the World Cup, World Grand Champions Cup, or the World League/World Grand Prix.

The competition formula of the FIVB World Championship has been constantly changed to fit the different number of teams that participate in each edition. The following rules usually apply:

Twenty-four teams participate in each event.

To participate in the event, a team must survive a number of qualification tournaments depending on its position in the FIVB World Rankings. Low-ranked teams may have to engage in up to three tournaments to be granted a berth; high-ranked teams typically play only one.

The competition is divided in two phases: a preliminary round and a final round. In the preliminary round, teams are organized in pools. Each team plays one match against all other teams in its pool.

When all the matches of the preliminary round have been played, the top 4 teams in each pool qualify for the following round(s), and the remaining ones leave the competition.

The FIVB has tried various different formats for the final round(s). For some years now (2004), there seems to be a consensus that at least semifinals and finals must be played according to the Olympic format.

Quarterfinals consist of groups of teams playing against each other.

The tournament implements very tight line-up restrictions: only twelve players are allowed, and no replacement is permitted, even in case of injuries.

The 2014 FIVB Volleyball Men's World Championship was held in Poland from 30 August to 21 September 2014. The tournament featured 24 teams to determine the world champions in men's volleyball. In addition to host nation Poland, 23 teams qualified for the tournament by means of continental and regional competitions. The matches took place in seven venues across seven Polish cities, with the final being held at the Spodek, Katowice.

We have result of the last match between any two teams attending World Championship and we want to predict the Champion. We assume the result of a match between any two teams would be the result of their last match against each other (the result of their last match before Championship).

Pool rankings

The team rankings per pool are determined by the points total after all matches have been played.

Match won 3-0 or 3-1: 3 points for the winner, 0 point for the loser.

Match won 3-2: 2 points for the winner, 1 point for the loser.

In case of tie, the teams will be classified according to the following criteria: number of matches won, points ratio and sets ratio

In the event of a tie at the end of each Round in the number of points scored by two or more teams, the teams will be classified according to the following criteria:

1. Number of Matches won.
2. Point ratio. Quotient resulting from: total number of points scored / total number of points lost.
3. Set ratio. If there is still a tie after applying the two criteria described above, the teams will be ranked by the quotient resulting from: total sets scored / total sets lost.



In the first round each team plays against each of the other 5 teams in its pool. After the end of the first round first 4 teams of each pool qualify for the next round. At the second round first 4 teams of pool A and first 4 teams of pool D are grouped together in pool E, and first 4 teams of pool B and first 4 teams of pool C are grouped together in pool F.

In the second round each team plays against each of the other 7 teams in its pool. After the end of the second round, first 3 teams of pool E and first 3 teams of pool F are qualified for the third round.

In the third round 6 teams that have qualified for this round are sorted according to pool standing procedure (they don't play against each other, they are just sorted according to their data from second round) and teams ranked 1, 4, 6 are grouped together in pool G and teams ranked 2, 3 and 5 are grouped together.

In the third round each team in pool G and pool H plays against two other teams. After the end of the third round first 2 teams of pool G and H are qualified for semi-final round. Team ranked 1 in pool G plays against team ranked 2 in pool H and team ranked 2 in pool G plays against team ranked 1 in pool H. The winner of these two games are qualified for the final game, and finally the champion will be revealed after they play against each other in the final game.

Input

First 4 lines of input describes 4 pools each consisting of 6 teams.

Names of each team is composed of lowercase English letters.

a1 a2 a3 a4 a5 a6

b1 b2 b3 b4 b5 b6

c1 c2 c3 c4 c5 c6

d1 d2 d3 d4 d5 d6

Next $(23 \times 24) / 2$ lines of input consist match results in this format.

teamA teamB setsWonByTeamA setsWonByTeamB resultOfTheFirstSet resultOfTheSecondSet ...

For example :

Iran Brasil 3-2 25-23 24-26 27-25 12-25 15-9

...

Output

Print the name of the Champion.

Examples

stdin	stdout
-------	--------

Note

Because of large input, input of this problem has been erased. Use your imagination!



Problem F. Huge Table

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

We have a rectangular table consisting 2^n rows and 2^m columns. The integer numbers are written from left to right, from top to bottom, in the cells of table. for an instance, when $n = 2$ and $m = 3$, our table is like this:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

For convenience of presentation we use the notation $[r, c]$ to denote a cell of table. $V[r, c]$ is the number written in $[r, c]$.

Let's define three functions:

- $popcount(n)$ is sum of digits in the binary representation of n . n should be non-negative. For example $popcount(7) = 3$ and $popcount(10) = 2$.
- $minpath(r, c)$: Consider a sub-rectangle of our table that $[1, 1]$ is its top-left corner and $[r, c]$ is its right-bottom. We want to go from $[1, 1]$ to $[r, c]$ and in each step we are just allowed to move one cell right or down. For each cell $[x, y]$ there is an entrance cost which is equal to $popcount(V[x, y])$. The function $minpath(r, c)$ calculates the minimum cost we should pay to reach from $[1, 1]$ to $[r, c]$.
- $sum(r_1, c_1, r_2, c_2)$ calculates $\sum popcount(V[r, c])$ for all cells like $[r, c]$ that lie in the sub-rectangle described by $[r_1, c_1]$ and $[r_2, c_2]$. Formally $(r_1 \leq r \leq r_2, c_1 \leq c \leq c_2)$.

Now you should find answer of the following interesting queries:

1. MINPATH r c : Print $minpath(r, c)$
2. SUM r_1 c_1 r_2 c_2 : Print $sum(r_1, c_1, r_2, c_2)$

Input

There are three integers in the first line: n , m and q ($0 \leq n \leq 50, 0 \leq m \leq 50, 1 \leq q \leq 100$).

The next q lines contain the queries in the format, given in the statement.

For MINPATH queries: $(1 \leq r \leq 2^n, 1 \leq c \leq 2^m)$ For SUM queries: $(1 \leq r_1 \leq r_2 \leq 2^n, 1 \leq c_1 \leq c_2 \leq 2^m)$

Output

For each query, output a single integer in a line corresponding to the answer of each query. Answers may become very large, so output them modulo 1000000009.



Examples

stdin	stdout
2 3 5	1
MINPATH 1 1	20
SUM 2 2 3 5	9
MINPATH 3 3	81
SUM 1 1 4 8	24
MINPATH 4 7	

Note

For the last query of the first sample “ MINPATH 4 7 ”, we can move in the following path:

$[1, 1] \rightarrow [1, 2] \rightarrow [1, 3] \rightarrow [1, 4] \rightarrow [2, 4] \rightarrow [3, 4] \rightarrow [3, 5] \rightarrow [3, 6] \rightarrow [4, 6] \rightarrow [4, 7]$

Cost of the path is: $popcount(1) + popcount(2) + popcount(3) + popcount(4) + popcount(12) + popcount(20) + popcount(21) + popcount(22) + popcount(30) + popcount(31) = 1 + 1 + 2 + 1 + 2 + 2 + 3 + 3 + 4 + 5 = 24$



Problem G. Coin Game

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 256 megabytes

Alice and Bob are very smart guys and they like to play all kinds of games in their spare time. The most amazing thing is that they always find the best strategy, and that's why they feel bored again and again. They just invented a new game, as they usually did.

They are playing with coins. They have a row of \$1 and \$2 coins. They want to change this row so that all \$1's are grouped together and all \$2's are grouped together. They are just allowed to swap two neighbor coins.

Using the best strategy what is the minimum number of swaps required to do this task?

Input

Each test case is a line with a string composed of 1 and 2.
($1 \leq \text{length of the input string} \leq 25000$)

Output

Print the minimum number of swaps required to do this task.

Examples

stdin	stdout
221212	3



Problem H. Dreams Were Important Too!

Input file: `stdin`
Output file: `stdout`
Time limit: 3 seconds
Memory limit: 256 megabytes

A few years ago, some silly guys stole a keyboard and an antiquated flag from a site. What do you think about the reaction of the administrator? It's obvious, he used all he could to return the flag and the keyboard. Whole country were searching for an entire month and many hypotheses had been proposed by funny people.

After a few years, another robbery happened. This time it was very complex, organized and supported by a big country. The robbers started their plan many years ago when we were limiting and rationing our site's capacity. Finally robbers succeeded to stole a "dream" from the site! This time the admin was surprised. After exchanging a few ineffective emails the poor stolen dream was forgotten and the admin said: "Instead of off topic discussions just suggest how you can help us for next year! We are too busy!" and real reasons were easily forgotten. "Finally the stolen dream remained just a dream!"

Let's get back to the story of the stolen flag. You should try to convince an unknown person that recreating a stolen triangular flag is not a hard task. You are given area and perimeter of the flag and you have to find integer coordinates of its vertices.

Input

The single line contains two positive space-separated integers $A2$ and P ($1 \leq A2 \leq 4 \times 10^4, 1 \leq P \leq 10^4$). $A2$ is twice the area and P is perimeter of the flag.

Output

If it's impossible to recreate the flag, output in the single line "Flag" (without quotes)
Otherwise output in the first line "Dream" (without quotes). In the next three lines output the coordinates of the triangle vertices. Each line should contain two space separated integers, each in range $[-10^6, 10^6]$.

Examples

stdin	stdout
12 12	Dream -1 2 2 6 2 2
15 15	Flag



Problem I. Coin Robbery

Input file: `stdin`
 Output file: `stdout`
 Time limit: 2 seconds
 Memory limit: 256 megabytes

Steve is a prominent coin collector and John is a professional thief, but Steve is not aware of this fact. They make friends and someday Steve asks John to help him move his coins to some other place. At the middle of the movement John plans a robbery and takes all Steve's coins. He has planned to go to airport and run away to Somalia as soon as possible.

This is the live conversation between Police station and Police officer.

Police Station: *We have found John's flight number, he is a mother lover, he wants to dedicate these coins to his mother in Somalia. he will use airport to run away to Somalia.*

Police Officer: *So ask airport security to arrest him.*

Police Station: *Right now we have no authority in airport. We have to do something in the city to arrest him out of airport or delay his arrival to the airport so he will miss his flight. John is smart, he has planned to arrive in airport and catch his plane with no delay in between.*

Police Officer: *We have some police officers in all intersections, we need to inform them to make block some intersections.*

We can not block starting intersection nor ending intersection (airport).

Intersections are positive integers in range $[0, n-1]$. Starting intersection of John is intersection 0 and airport is intersection $n-1$. There maybe several streets between two intersections.

Now it's your turn, inform minimum number of police officers at intersections so that we can arrest John in one of the intersections or we can delay his arrival to airport so that he will miss his flight.

Input

First line of input contains two integers, n and m , the number of intersections and the number of streets, correspondingly.

Each of the next m lines contains a description of a bidirectional street, two ends of the street and the length of the street.

$(2 \leq n \leq 100)$

$(1 \leq m \leq 10000)$

$(1 \leq \text{length of each street} \leq 100)$

Output

If it's impossible to block John from entering airport, print IMPOSSIBLE, otherwise print the minimum number of intersections we need to block.

Examples

stdin	stdout
3 3 0 1 1 1 2 1 0 2 3	1



Problem J. Bimetallic coins

Input file: stdin
Output file: stdout
Time limit: 2 seconds
Memory limit: 256 megabytes

King Coin IV loves bi-metallic coins. Bi-metallic coins are coins consisting of two different alloys, arranged with an outer ring around a contrasting center. After consulting with the most prominent metallurgists, the king decided to abolish simple coins and increase the use of bi-metallic coins in his country.



They have n types of metals. Their technology allows them to mix two different basic metals in equal ratio and produce an alloy. Experienced metallurgists discovered that each alloy has a different thermal area expansion coefficient. For two different alloys, these coefficients help them to determine which one can be used as outer ring in a bi-metallic coin. An alloy can be used as outer ring if its thermal coefficient is less than thermal coefficient of inner part. For each alloy they know the costs to create an outer ring and inner part using that alloy. Because it's very expensive to make a revolution in mediums of exchanges in a country, the king wants to have maximum types of bi-metallic coins with minimum production cost. Note that each alloy can not be used in more than one type of bi-metallic coin.

Input

First line contains an integer n ($3 \leq n \leq 50$) — the number of basic metals.

Following $3n$ lines describe three $n \times n$ matrices C, I, O .

C consists of real numbers with exactly three digits after the decimal point. $C_{i,j}$ is thermal area expansion coefficient of alloy composed by metals i and j . ($0.000 < C_{i,j} < 1.000$)

I consists of integers. $I_{i,j}$ is cost of creating an inner part from alloy composed by metals i and j . ($1 \leq I_{i,j} \leq 10000$)

O consists of integers. $O_{i,j}$ is cost of creating an outer ring from alloy composed by metals i and j . ($1 \leq O_{i,j} \leq 10000$)

C, I, O are symmetric and their diagonals are always zero.

Output

Output two space separated integers in a single line. The maximum number of bi-metallic coin types and the minimum cost of producing these types while each alloy can not be used in more than one type.

Examples

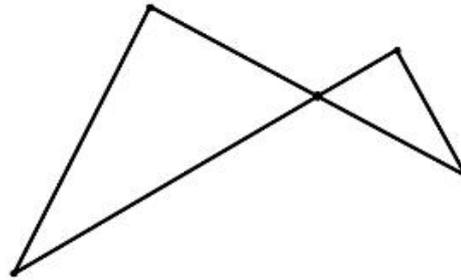
stdin	stdout
<pre>3 0.000 0.012 0.312 0.012 0.000 0.111 0.312 0.111 0.000 0 3 5 3 0 4 5 4 0 0 4 9 4 0 5 9 5 0</pre>	<pre>1 8</pre>



Problem K. Censorship!

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

You are so lucky because the story of this problem which was about cheating has been censored by the manager of the contest, and you don't have to read that long story! Given n points in a $2D$ plane, You have to use these points to make a magical shape like a bow tie. Our bow tie like shape consists of two triangles that share a vertex. We call it center point. Now you may want to know a little more about this bow tie like shape. As you can see in the picture this shape is made up of 5 points and 6 segments that make two triangles. Four of these six segments must share the center point as a point and these segments must be in two lines that intersect each other in center point. One another important condition to make a bow tie like shape is that no other points are allowed to be inside or on the segments of the shape. In this problem you must count the number of different bow ties like shapes that can be drawn using the points.



Input

The first line of input is the number of points in plane, n ($5 \leq n \leq 500$). In the next lines there are n lines with two integer numbers that are coordinates of points, x_i and y_i ($-10^4 \leq x_i, y_i \leq 10^4$). The points are different, but clearly some points maybe collinear.

Output

Output the number of different bow tie like shapes that we can make using this set of points.

Examples

stdin	stdout
<pre>5 0 0 1 0 0 1 -1 0 0 -1</pre>	2
<pre>7 1 1 4 -2 2 2 0 2 -1 -1 1 0 1 2</pre>	5