

A. Cinema

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The capital of Berland has the only movie theater in the country. Besides, it consists of only one room. The room is divided into n rows, each row consists of m seats.

There are k people lined up to the box office, each person wants to buy exactly one ticket for his own entertainment. Before the box office started selling tickets, each person found the seat that seemed best for him and remembered it as a pair of coordinates (x_i, y_i) , where x_i is the row number, and y_i is the seat number in this row.

It is possible that some people have chosen the same place, then when some people see their favorite seat taken in the plan of empty seats in the theater, they choose and buy a ticket to another place. Each of them has the following logic: let's assume that he originally wanted to buy a ticket to seat (x_1, y_1) , then when he comes to the box office, he chooses such empty seat (x_2, y_2) , which satisfies the following conditions:

- the value of $|x_1 - x_2| + |y_1 - y_2|$ is minimum
- if the choice is not unique, then among the seats that satisfy the first condition, this person selects the one for which the value of x_2 is minimum
- if the choice is still not unique, among the seats that satisfy the first and second conditions, this person selects the one for which the value of y_2 is minimum

Your task is to find the coordinates of a seat for each person.

Input

The first input line contains three integers n, m, k ($1 \leq n, m \leq 2000, 1 \leq k \leq \min(n \cdot m, 10^5)$) — the number of rows in the room, the number of seats in each row and the number of people in the line, correspondingly. Each of the next k lines contains two integers x_i, y_i ($1 \leq x_i \leq n, 1 \leq y_i \leq m$) — the coordinates of the seat each person has chosen. Numbers on the same line are separated by a space. The pairs of coordinates are located in the order, in which people stand in the line, starting from the head (the first person in the line who stands in front of the box office) to the tail (the last person in the line).

Output

Print k lines, each containing a pair of integers. Print on the i -th line x_i, y_i — the coordinates of the seat, for which the person who stands i -th in the line will buy the ticket.

Examples

[illegible]

2 2
2 2
2 2

output

2 2
1 2
2 1
2 3
3 2
1 1
1 3
3 1
3 3
4 2
4 1
4 3

B. Drinks

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

Little Vasya loves orange juice very much. That's why any food and drink in his kitchen necessarily contains orange juice. There are n drinks in his fridge, the volume fraction of orange juice in the i -th drink equals p_i percent.

One day Vasya decided to make himself an orange cocktail. He took equal proportions of each of the n drinks and mixed them. Then he wondered, how much orange juice the cocktail has.

Find the volume fraction of orange juice in the final drink.

Input

The first input line contains a single integer n ($1 \leq n \leq 100$) — the number of orange-containing drinks in Vasya's fridge. The second line contains n integers p_i ($0 \leq p_i \leq 100$) — the volume fraction of orange juice in the i -th drink, in percent. The numbers are separated by a space.

Output

Print the volume fraction in percent of orange juice in Vasya's cocktail. The answer will be considered correct if the absolute or relative error does not exceed 10^{-4} .

Examples

input
3 50 50 100
output
66.666666666667

input
4 0 25 50 75
output
37.500000000000

Note

Note to the first sample: let's assume that Vasya takes X milliliters of each drink from the fridge. Then the volume of pure juice in the cocktail will equal $\frac{x}{2} + \frac{x}{2} + x = 2 \cdot x$ milliliters. The total cocktail's volume equals $3 \cdot X$ milliliters, so the volume fraction of the juice in the cocktail equals $\frac{2x}{3x} = \frac{2}{3}$, that is, **66.(6)** percent.

C. Football Championship

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Any resemblance to any real championship and sport is accidental.

The Berland National team takes part in the local Football championship which now has a group stage. Let's describe the formal rules of the local championship:

- the team that kicked most balls in the enemy's goal area wins the game;
- the victory gives 3 point to the team, the draw gives 1 point and the defeat gives 0 points;
- a group consists of four teams, the teams are ranked by the results of six games: each team plays exactly once with each other team;
- the teams that get places 1 and 2 in the group stage results, go to the next stage of the championship.

In the group stage the team's place is defined by the total number of scored points: the more points, the higher the place is. If two or more teams have the same number of points, then the following criteria are used (the criteria are listed in the order of falling priority, starting from the most important one):

- the difference between the total number of scored goals and the total number of missed goals in the championship: the team with a higher value gets a higher place;
- the total number of scored goals in the championship: the team with a higher value gets a higher place;
- the lexicographical order of the name of the teams' countries: the country with the lexicographically smaller name gets a higher place.

The Berland team plays in the group where the results of 5 out of 6 games are already known. To be exact, there is the last game left. There the Berland national team plays with some other team. The coach asks you to find such score $X:Y$ (where X is the number of goals Berland scored and Y is the number of goals the opponent scored in the game), that fulfills the following conditions:

- $X > Y$, that is, Berland is going to win this game;
- after the game Berland gets the 1st or the 2nd place in the group;
- if there are multiple variants, you should choose such score $X:Y$, where value $X - Y$ is minimum;
- if it is still impossible to come up with one score, you should choose the score where value Y (the number of goals Berland misses) is minimum.

Input

The input has five lines.

Each line describes a game as " $team_1 team_2 goals_1:goals_2$ " (without the quotes), what means that team $team_1$ played a game with team $team_2$, besides, $team_1$ scored $goals_1$ goals and $team_2$ scored $goals_2$ goals. The names of teams $team_1$ and $team_2$ are non-empty strings, consisting of uppercase English letters, with length of no more than 20 characters; $goals_1, goals_2$ are integers from 0 to 9.

The Berland team is called "BERLAND". It is guaranteed that the Berland team and one more team played exactly 2 games and the other teams played exactly 3 games.

Output

Print the required score in the last game as $X:Y$, where X is the number of goals Berland scored and Y is the number of goals the opponent scored. If the Berland team does not get the first or the second place in the group, whatever this game's score is, then print on a single line "IMPOSSIBLE" (without the quotes).

Note, that the result score can be very huge, 10:0 for example.

Examples

input
AERLAND DERLAND 2:1 DERLAND CERLAND 0:3 CERLAND AERLAND 0:1 AERLAND BERLAND 2:0 DERLAND BERLAND 4:0
output
6:0

input
AERLAND DERLAND 2:2 DERLAND CERLAND 2:3 CERLAND AERLAND 1:3

AERLAND BERLAND 2:1
DERLAND BERLAND 4:1

output

IMPOSSIBLE

Note

In the first sample "BERLAND" plays the last game with team "CERLAND". If Berland wins with score 6:0, the results' table looks like that in the end:

- 1. AERLAND (points: 9, the difference between scored and missed goals: 4, scored goals: 5)
- 2. BERLAND (points: 3, the difference between scored and missed goals: 0, scored goals: 6)
- 3. DERLAND (points: 3, the difference between scored and missed goals: 0, scored goals: 5)
- 4. CERLAND (points: 3, the difference between scored and missed goals: -4, scored goals: 3)

In the second sample teams "AERLAND" and "DERLAND" have already won 7 and 4 points, respectively. The Berland team wins only 3 points, which is not enough to advance to the next championship stage.

D. Programming Language

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Recently, Valery have come across an entirely new programming language. Most of all the language attracted him with template functions and procedures. Let us remind you that templates are tools of a language, designed to encode generic algorithms, without reference to some parameters (e.g., data types, buffer sizes, default values).

Valery decided to examine template procedures in this language in more detail. The description of a template procedure consists of the procedure name and the list of its parameter types. The generic type `T` parameters can be used as parameters of template procedures.

A procedure call consists of a procedure name and a list of variable parameters. Let's call a procedure *suitable* for this call if the following conditions are fulfilled:

- its name equals to the name of the called procedure;
- the number of its parameters equals to the number of parameters of the procedure call;
- the types of variables in the procedure call match the corresponding types of its parameters. The variable type matches the type of a parameter if the parameter has a generic type `T` or the type of the variable and the parameter are the same.

You are given a description of some set of template procedures. You are also given a list of variables used in the program, as well as direct procedure calls that use the described variables. For each call you need to count the number of procedures that are suitable for this call.

Input

The first line contains a single integer n ($1 \leq n \leq 1000$) — the number of template procedures. The next n lines contain the description of the procedures specified in the following format:

"void procedureName (type_1, type_2, ..., type_t)" ($1 \leq t \leq 5$), where void is the keyword, procedureName is the procedure name, type_i is the type of the next parameter. Types of language parameters can be "int", "string", "double", and the keyword "T", which denotes the generic type.

The next line contains a single integer m ($1 \leq m \leq 1000$) — the number of used variables. Next m lines specify the description of the variables in the following format:

"type variableName", where type is the type of variable that can take values "int", "string", "double", variableName — the name of the variable.

The next line contains a single integer k ($1 \leq k \leq 1000$) — the number of procedure calls. Next k lines specify the procedure calls in the following format:

"procedureName (var_1, var_2, ..., var_t)" ($1 \leq t \leq 5$), where procedureName is the name of the procedure, var_i is the name of a variable.

The lines describing the variables, template procedures and their calls may contain spaces at the beginning of the line and at the end of the line, before and after the brackets and commas. Spaces may be before and after keyword void. The length of each input line does not exceed 100 characters. The names of variables and procedures are non-empty strings of lowercase English letters and numbers with lengths of not more than 10 characters. Note that this is the only condition at the names. Only the specified variables are used in procedure calls. The names of the variables are distinct. No two procedures are the same. Two procedures are the same, if they have identical names and identical ordered sets of types of their parameters.

Output

On each of k lines print a single number, where the i -th number stands for the number of suitable template procedures for the i -th call.

Examples

input

```
4
void f(int,T)
void f(T, T)
void foo123 ( int, double, string,string )
void p(T,double)
3
int a
string s
double x123
5
f(a, a)
f(s,a )
foo (a,s,s)
f ( s ,x123)
proc(a)
```

output
2 1 0 1 0

input
6 void f(string,double,int) void f(int) void f (T) void procedure(int,double) void f (T, double,int) void f(string, T,T) 4 int a int x string t double val 5 f(t, a, a) f(t,val,a) f(val,a, val) solve300(val, val) f (x)

output
1 3 0 0 2

E. Tractor College

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

While most students still sit their exams, the tractor college has completed the summer exam session. In fact, students study only one subject at this college — the Art of Operating a Tractor. Therefore, at the end of a term a student gets only one mark, a three (satisfactory), a four (good) or a five (excellent). Those who score lower marks are unfortunately expelled.

The college has n students, and oddly enough, each of them can be on scholarship. The size of the scholarships varies each term. Since the end-of-the-term exam has just ended, it's time to determine the size of the scholarship to the end of next term.

The monthly budget for the scholarships of the Tractor college is S rubles. To distribute the budget optimally, you must follow these rules:

- The students who received the same mark for the exam, should receive the same scholarship;
- Let us denote the size of the scholarship (in roubles) for students who have received marks 3, 4 and 5 for the exam, as k_3 , k_4 and k_5 , respectively. The values k_3 , k_4 and k_5 must be integers and satisfy the inequalities $0 \leq k_3 \leq k_4 \leq k_5$;
- Let's assume that c_3 , c_4 , c_5 show how many students received marks 3, 4 and 5 for the exam, respectively. The budget of the scholarship should be fully spent on them, that is, $c_3 \cdot k_3 + c_4 \cdot k_4 + c_5 \cdot k_5 = S$;
- Let's introduce function $f(k_3, k_4, k_5) = |c_3 \cdot k_3 - c_4 \cdot k_4| + |c_4 \cdot k_4 - c_5 \cdot k_5|$ — the value that shows how well the scholarships are distributed between students. In the optimal distribution function $f(k_3, k_4, k_5)$ takes the **minimum** possible value.

Given the results of the exam, and the budget size S , you have to find the optimal distribution of the scholarship.

Input

The first line has two integers n , S ($3 \leq n \leq 300$, $1 \leq S \leq 3 \cdot 10^5$) — the number of students and the budget size for the scholarship, respectively. The second line contains n integers, where the i -th number represents the mark that the i -th student got for the exam. It is guaranteed that at each mark was given to at least one student.

Output

On a single line print three integers k_3 , k_4 and k_5 — the sought values that represent the optimal distribution of the scholarships. If there are multiple optimal answers, print any of them. If there is no answer, print -1.

Examples

input
5 11 3 4 3 5 5
output
1 3 3

input
6 15 5 3 3 4 4 5
output
-1