Hong Kong Olympiad in Informatics 2024/25 Junior Group

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
J251	Spelling Bee	1.000 s	256 MB	21 + 22 + 7 + 23 + 18 + 9
J252	Dual-Election	1.500 s	256 MB	19 + 22 + 21 + 25 + 13
J253	Tournament II	1.000 s	256 MB	11 + 16 + 39 + 12 + 22
J254	Wandering Around the Garden	1.000 s	256 MB	4 + 6 + 20 + 10 + 25 + 20 + 15

Notice:
Unless otherwise specified, inputs and outputs shall follow the format below:

- One space between a number and another number or character in the same line.
- No space between characters in the same line.
- Each string shall be placed in its own separate line.
- Outputs will be automatically fixed as follows: Trailing spaces in each line will be removed and an end-of-line character will be added to the end of the output if not present. All other format errors will not be fixed.

C++ programmers should be aware that using C++ streams (`cin` / `cout`) may lead to I/O bottlenecks and substantially lower performance.

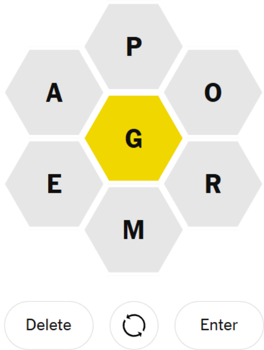
For some problems 64-bit integers may be required. In C++ it is `long long` and its token for `scanf`/`printf` is `%lld`.

All tasks are divided into subtasks. You need to pass all test cases in a subtask to get points.

J251 - SPELLING BEE

Time Limit: 1.000 s / Memory Limit: 256 MB

Alice is now playing the game *Spelling Bee*. The game presents players with 7 different letters arranged in a hexagonal grid, and the player scores points by forming a word using some or all of those letters. The player can **only** use those letters, but each letter can be used more than once.



An example of the hexagonal grid in the game.

The letter in the center of the hexagonal grid is called the *center letter*, and Alice must include this letter in the word she forms. If Alice's word does not include the center letter, the word would be rejected with the reason `NO CENTER LETTER`. Besides, the word must consist of at least 4 letters, otherwise it would be rejected with the reason `TOO SHORT`. Note that if a word fails both checks (that is, the word is both too short and does not include the center letter), it will still be rejected with the reason `TOO SHORT`.

If the word Alice forms satisfies the two constraints above, the word will be accepted and she will obtain a score computed as follows:

- If the word consists of 4 letters, she gets 1 point.
- Otherwise, if the word consists of at least 5 letters, she gets a score equivalent to the number of letters in the word.
- A bonus 7 points will be added to the score if the word she forms is a *pangram*. The word is a pangram if and only if it uses all 7 letters in the hexagonal grid at least once.

Alice has just come up with a word. Before inputting the word into the game, she wonders what her score will be. Can you write a checker for her to calculate the score of the word, or determine if the word is invalid?

INPUT

The first line of input consists of a single uppercase letter C, the center letter in the hexagonal grid.

The second line of input consists of a string H of 6 uppercase letters, the other letters in the hexagonal grid. It is guaranteed that all 7 letters in the hexagonal grid are distinct.

The third line of input consists of a string W of uppercase letters, representing Alice's word. Alice's word only consists of the letters in the hexagonal grid. It is not guaranteed that Alice's word is a valid English word, and you are not required to validate this.

OUTPUT

If the word is rejected, output the reason of rejection, which should either be `TOO SHORT` or `NO CENTER LETTER`, the checking is case-insensitive.

Otherwise, output a single integer: the score (in points) of the word computed according to the rules stated above.

SAMPLE TESTS

	Input	Output
1	G PORMEA PROGRAMMER	17

The word consists of 10 letters, and uses all 7 letters in the hexagonal grid at least once. Therefore, it scores $10 + 7 = 17$ points.

	Input	Output
2	G PORMEA GAME	1

The word consists of 4 letters and it is not a pangram (it does not consist of the letter P, for instance). Therefore, it scores $1 + 0 = 1$ point.

	Input	Output
3	E POCRNS SOONER	6
4	Z SPELIN EELS	NO CENTER LETTER
5	B DONUTS BUN	TOO SHORT
6	S PIRATE PI	TOO SHORT
7	E SALMON SEEM	1

SUBTASKS

For all cases:
C is an uppercase letter
H consists of exactly 6 uppercase letters
W consists of 2 to 13 (inclusive) uppercase letters
W only consists of the letters in C and H

	Points	Constraints
1	21	Alice's word is not a pangram Alice's word is accepted
2	22	Alice's word is not a pangram Alice's word consists of at least 5 letters
3	7	Alice's word is not a pangram
4	23	Alice's word is accepted
5	18	Alice's word consists of at least 5 letters
6	9	No additional constraints

J252 - DUAL-ELECTION

Time Limit: 1.500 s / Memory Limit: 256 MB

In Byteland Secondary School, each class organize an election to select a class president every year. However, there can be many candidates in a class and only one candidate can become the class president. As a result, many students are unsatisfied with the election results when their supported candidate loses the election in the end.

This year, Byteland Secondary School introduces a dual-election scheme, which selects **two** class presidents from each class instead.

There are $N + M$ students in the class, of whom N students are the voters and the other M students are the candidates. The voters are numbered from $1, 2, \ldots, N$ and the candidates are numbered from $1, 2, \ldots, M$. Each voter votes for **two** supported candidates. Formally, the i-th voter votes for the candidates X_i and Y_i, where $1 \leq X_i, Y_i \leq M$ and $X_i \neq Y_i$.

After all voters cast their votes, two winners would be chosen. A voter is satisfied with the results of the election if and only if the voter voted for at least one of the winners in the election. As the class teacher, you have the authority to determine the winners of the election, regardless of which candidates receive the most votes. Therefore, you decide to select two class presidents (i.e., the election winners) in a way that maximizes the number of satisfied voters. Can you find out which two candidates you should select?

INPUT

The first line of input consists of two integers N and M, the number of voters and candidates respectively.

The next N lines of input each contains two integers X_i and Y_i, representing the two candidates that voter i voted for.

OUTPUT

Output two integers, the indices of the two winners of the election, that maximize the number of satisfied voters. The two integers must be two **different** numbers from 1 to M.

If there are multiple ways to pick the winners of the election such that the number of satisfied voters is maximized, you may output any of them. You may also output the two winners in any order.

SAMPLE TESTS

1

Input	Output
5 4 1 2 3 4 2 3 1 4 1 4	2 4

By selecting the candidates 2 and 4 as the winners, all voters will be satisfied since

- Voter 1 voted for candidate 2 who wins the election.
- Voter 2 voted for candidate 4 who wins the election.
- Voter 3 voted for candidate 2 who wins the election.
- Voter 4 voted for candidate 4 who wins the election.
- Voter 5 voted for candidate 4 who wins the election.

Besides `2 4`, the answers `4 2`, `1 3` and `3 1` will also be accepted.
This sample satisfies the constraints of Subtasks 1, 2 and 5.

	Input	Output
2	3 4 1 4 2 4 3 4	1 4

By selecting the candidates 1 and 4 as the winners, all voters will be satisfied since

- Voter 1 voted for candidate 1 (and candidate 4 at the same time) who wins the election.
- Voter 2 voted for candidate 4 who wins the election.
- Voter 3 voted for candidate 4 who wins the election.

This sample satisfies the constraints of all subtasks.

	Input	Output
3	3 6 1 4 2 5 3 6	1 3

This sample satisfies the constraints of all subtasks.

	Input	Output
4	3 5 1 5 2 5 3 4	3 5

This sample satisfies the constraints of all subtasks.

SUBTASKS

For all cases:
$1 \leq N \leq 100000$
$2 \leq M \leq 200000$
$1 \leq X_i, Y_i \leq M$ for $1 \leq i \leq N$
$X_i \neq Y_i$ for $1 \leq i \leq N$

	Points	Constraints
1	19	$1 \leq N \leq 100$ $2 \leq M \leq 200$
2	22	$2 \leq M \leq 1000$
3	21	$M > N$ $X_i = i$ for $1 \leq i \leq N$ $Y_i > N$ for $1 \leq i \leq N$
4	25	Each voter votes for a distinct pair of candidates Formally, $(X_i, Y_i) \neq (X_j, Y_j)$ and $(X_i, Y_i) \neq (Y_j, X_j)$ for $1 \leq i < j \leq N$
5	13	No additional constraints

J253 - TOURNAMENT II

Time Limit: 1.000 s / Memory Limit: 256 MB

Byteland is hosting an exciting typing speed tournament, and you are the organizer! The tournament includes N contestants and consists of $N-1$ rounds, with each round eliminating exactly one contestant. The last remaining contestant at the end of the final round will be declared the champion. For simplicity, contestants are indexed from 1 to N.

In each round, **all** contestants that have not been eliminated would type a specific passage, and their typing speeds in that round are measured. The contestant with the lowest typing speed in that round is eliminated from the tournament. If multiple contestants share the same lowest typing speed, the contestant with the smallest index among the contestants sharing the lowest typing speed in this round is eliminated.

Due to logistical constraints, contestants are participating in the tournament virtually through an online system. The system records the tournament results by saving the typing speeds for each round in which a contestant participated. Specifically, the database saves a list of records, with each record having the form (r, c, s) which represents contestant c participated in round r and achieved a typing speed of s. Note that there are totally $\left(\frac{N(N+1)}{2} - 1\right)$ records in the database.

Using this information, you expected to reconstruct the tournament results afterwards. However, upon examining the database after the tournament, you realize that the round information was not saved due to a system bug! Formally, you now only have a list of records (C_i, S_i) where $1 \le i \le \frac{N(N+1)}{2} - 1$, representing contestant C_i participated in an unspecified round and achieved a typing speed of S_i.

Despite this limitation, you still need to issue certificates to the contestants. Therefore, you need to know what might have happened in the tournament. Specifically, you must find the typing speeds of the contestants in each round, that are consistent with the tournament rules and your records. However, there might be more bugs in the system that makes it impossible to construct consistent typing speeds! If that is the case, report it.

INPUT

The first line of the input contains an integer N.
The i-th line of the following $\left(\frac{N(N+1)}{2} - 1\right)$ lines each contains two integers C_i and S_i, indicating that contestant C_i achieved a typing speed of S_i in an unspecified round.

OUTPUT

If no tournament results are consistent with the given records, output `oh no`.
Otherwise, output N lines, each containing $N-1$ integers.
On the i-th line, the j-th integer should be the typing speed achieved by contestant i in round j, or -1 if contestant i has been eliminated by that round.
If there are multiple possible tournament results, output any of them.

SAMPLE TESTS

1

Input	Output
3 1 95 1 30 2 60 2 60 3 55	95 30 60 60 55 -1

The output describes a possible configuration of the contestants' typing speeds. In particular:

- In round one, contestants 1, 2 and 3 type with speeds 95, 60 and 55 respectively. Contestant 3 is the slowest among the three contestants, and hence is eliminated.
- In round two, contestants 1 and 2 type with speeds 30 and 60 respectively. Contestant 1 is the slowest among the two contestants, and hence is eliminated.

2

Input	Output
4 1 20 2 24 2 20 3 25 3 53 3 20 4 82 4 21 4 18	20 -1 -1 20 24 -1 20 25 53 21 82 18

The output describes a possible configuration of the contestants' typing speeds. In particular:

- In round one, contestants 1, 2, 3 and 4 type with speeds 20, 20, 20 and 21 respectively. Contestants 1, 2 and 3 share the slowest typing speed in this round.
  Contestant 1 is eliminated because the index 1 is the smallest index among these three contestants.
- In round two, contestants 2, 3 and 4 type with speeds 24, 25 and 82 respectively. Contestant 2 is the slowest among the three contestants, and hence is eliminated.
- In round three, contestants 3 and 4 type with speeds 53 and 18 respectively. Contestant 4 is the slowest among the two contestants, and hence is eliminated.

3

Input	Output
3 1 1 1 3 2 5 2 2 2 4	oh no

The database somehow contains three records for contestant 2 and no records for contestant 3. A bug must have occurred and no valid typing speed configurations can be constructed.

4

Input	Output
3 1 5 2 1 2 2 3 4 3 3	oh no

SUBTASKS

For all cases:
$2 \leq N \leq 1200$
$1 \leq C_i \leq N$ for $1 \leq i \leq \frac{N(N+1)}{2} - 1$
$0 \leq S_i \leq 10^7$ for $1 \leq i \leq \frac{N(N+1)}{2} - 1$

	Points	Constraints
1	11	$N = 3$ $C_1 = 1$ $C_2 = C_3 = 2$ $C_4 = C_5 = 3$
2	16	$N \leq 300$ $S_i = S_j$ if $C_i = C_j$ for $1 \leq i, j \leq \frac{N(N+1)}{2} - 1$
3	39	$N \leq 300$ $S_i \neq S_j$ if $i \neq j$ for $1 \leq i, j \leq \frac{N(N+1)}{2} - 1$
4	12	$N \leq 300$
5	22	No additional constraints

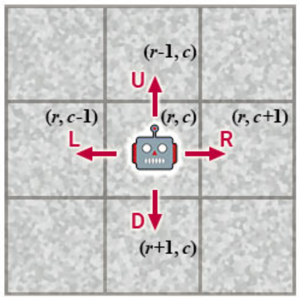
J254 - WANDERING AROUND THE GARDEN

Time Limit: 1.000 s / Memory Limit: 256 MB

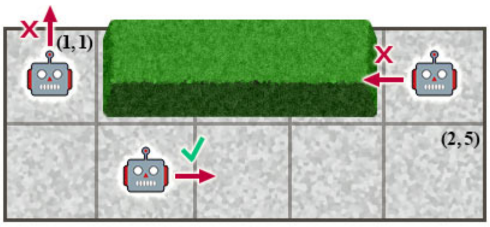
Alice is visiting a garden. The garden is narrow and may be regarded as a $2 \times N$ rectangular grid. The coordinates of the square at the i-th row and the j-th column is (i, j), where $1 \leq i \leq 2$ and $1 \leq j \leq N$. Currently, there are K groups of bushes in the garden, where each group is a $1 \times W$ subgrid, such that there are bushes planted at each square in the subgrid. It is guaranteed that $K \leq 2$, so that **there are at most two groups of bushes in the garden**. It is also guaranteed that there are no bushes on the first and last columns.

Alice would like to wander around the garden, but she is feeling tired and therefore would like her robot, Robo, to do the wandering for her. Starting at $(1, 1)$, the goal is for Robo to make **exactly** T moves and to first visit (G, N) after T moves. Going one square up, down, left, or right is considered one move, and these are the only allowable moves. In terms of coordinates, if Robo is currently at (r, c):

- after going one square up, its new coordinates are $(r - 1, c)$;
- after going one square down, its new coordinates are $(r + 1, c)$;
- after going one square left, its new coordinates are $(r, c - 1)$;
- after going one square right, its new coordinates are $(r, c + 1)$.



If a move leads to Robo either leaving the garden or hitting the bushes (for the bushes will trap Robo), it is considered invalid; otherwise it is valid.



Alice is so tired, however, that even writing the sequence of moves for Robo may be too tiring for her. Fortunately, Robo comes with a special operation `()r`, which repeats the command in the brackets (so, executed twice in total), therefore allowing Alice to write a potentially shorter command. Here are the formal rules concerning how to write a command:

- `L`, `R`, `U`, and `D` are commands, which instructs Robo to move one square in the left, right, up, and down directions, respectively;
- If C_1 and C_2 are commands, then C_1C_2 is a command which instructs Robo to execute C_1 then C_2;
- If C is a command, then $(C)r$ is a command, which instructs Robo to execute C twice.

For example, the command `R(L(R)r)rD` expands to `RLRRLRRD`, so Robo will make eight moves according to the expanded command (assuming that all moves are valid). Note that the `()r` operation may be nested.

A command is said to be valid if and only if Robo only makes valid moves by following the command. Given the garden configuration, the target square, and the desired number of moves T, write a valid short command so that Robo makes exactly T moves and first visits the target square after T moves, or report that it is impossible. That is, output a command so that:

- Robo makes exactly T moves;
- Each of the T moves are valid, i.e. they will not lead Robo out of the garden boundaries or into the bushes;
- Robo reaches (G, N) after T moves;
- Robo does not visit (G, N) before the T-th move; and
- The length of the command (the number of characters that the command contains) is sufficiently short (see "Scoring").

INPUT

The first line of input consists of three integers N, T, G, the number of columns of the garden, the desired number of moves, and the row number of the target square.

The second line of input consists of an integer K, the number of groups of bushes in the garden.

K lines follow. The i-th of these K lines consists of three integers R_i, A_i, B_i, meaning that the i-th group of bushes occupies the squares $(R_i, A_i), (R_i, A_i + 1), \ldots, (R_i, B_i)$.

OUTPUT

If there is no valid command so that Robo makes exactly T moves and first visits the target square (G, N) after T moves, output `Impossible`. Otherwise, output a valid command in one line. Your score will depend on the length of the command (see "Scoring").

SCORING

Within a subtask, if for at least one test case, your program either fails to find a solution when one exists, or fails to report an impossible case, you score 0% in the subtask. Otherwise, let L be the maximum length of the output command over all test cases for which a solution exists.

- If $L \leq 320$, you score 100% in the subtask.
- If $320 < L \leq 7000$, you score 60% in the subtask.
- If $7000 < L \leq 100000$, you score 35% in the subtask.
- If $L > 100000$, you score 0% in the subtask.

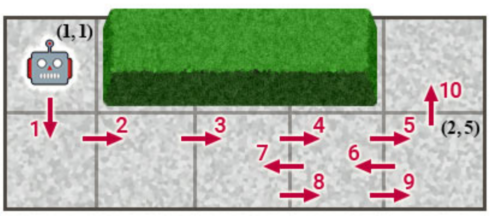
It can be shown that if there exista a solution, there also exists at least one solution such that $L \leq 320$. If there are several possible solutions, output any of them.

SAMPLE TESTS

	Input	Output
1	5 10 1 1 1 2 4	D((R)r)rLLRRU

The output command expands to `DRRRRLLRRU`, which is valid. Robo makes exactly 10 moves, reaching the target square $(1, 5)$ at the end, but not earlier. So, all the solution requirements are satisfied. The length of the command is 13, which scores 100% of the points.

Some examples of invalid solutions include `DUDUDURRRR` (hitting the bushes), `UUDDDRRRRU` (going out of bounds), `D((R)r)rLLRRUD` (too many moves), and `D((R)r)rUDUDU` (first visit of $(1, 5)$ is too early).



	Input	Output
2	5 10 1 2 1 2 4 2 2 4	Impossible
3	5 10 2 0	Impossible

SUBTASKS

For all cases:
$3 \leq N \leq 10^7$
$1 \leq T \leq 10^7$
$1 \leq G \leq 2$
$0 \leq K \leq 2$
$1 \leq R_i \leq 2$ for $1 \leq i \leq K$
$2 \leq A_i \leq B_i \leq N-1$ for $1 \leq i \leq K$
If there are two groups of bushes, they do not overlap.

	Points	Constraints
1	4	$N \leq 300$ $T = N-1$ $G = 1$ $K = 0$
2	6	$N \leq 300$ $T = N-1$
3	20	$N \leq 300$ $T \leq 320$ $K \leq 1$
4	10	$N \leq 300$ $T \leq 320$
5	25	$K = 0$
6	20	$K \leq 1$
7	15	No additional constraints