Hong Kong Olympiad in Informatics 2016/17 Junior Group

Task Overview

ID	Name	Time Limit	Memory Limit	Subtasks
J171	Acronym	1.000 s	256 MB	15 + 16 + 17 + 14 + 10 + 28
J172	Card Game	1.000 s	256 MB	17 + 11 + 9 + 20 + 14 + 29
J173	Fibonacci Word	1.000 s	256 MB	3 + 24 + 17 + 19 + 37
J174	Crosses	1.000 s	256 MB	18 + 12 + 13 + 15 + 9 + 11 + 22

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 Pascal it is `int64`. In C/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.

J171 - ACRONYM

Time Limit: 1.000 s / Memory Limit: 256 MB

In daily life, we encounter many lengthy phrases, many of which are difficult to memorise. To memorise these phrases more easily and to save time, we often use acronyms to refer to these phrases. For example, `HKOI` is an acronym for `Hong Kong Olympiad in Informatics`.

Sometimes, the use of acronyms may cause problems. One such problem is that different phrases may share the same acronym. For example, if you google `IMO`, both `International Mathematical Olympiad` and `International Maritime Organization` come out as top hits.

Because of this problem, one may have to guess the correct phrase. Given an acronym, which is a string consisting of capital English letters, and N guesses, your task is to determine whether the guesses are reasonable.

Each guess is a phrase. A phrase consists of one or more words. Each word is a string of English letters. In a phrase, words are separated by exactly one space.

A guess is reasonable if and only if it is possible to choose the first letters of several words of the phrase to form the acronym. For example, for the acronym `HKOI`, `Hong Kong Olympiad in Informatics` is a reasonable guess, so are `hOnG KONG a b c olymPica informatica`, `h k o i`, and `honorable KING of ippyland`. `HKOI` and `k h o i` are not reasonable guesses.

If a guess is reasonable, output a standard form of it. A phrase is in standard form if it satisfies several constraints:

1. For the chosen words, their first letters are capitalised.
2. All other letters are small letters.
3. The capital letters form the acronym.

For example, for the acronym `HKOI` and the guess `hOnG KONG olimpiada en informatica`, your program should output `Hong Kong Olimpiada en Informatica`.

INPUT

The first line consists of a string: the acronym.

The second line contains an integer N, the number of guesses to follow.

The next N lines of input each consists of a guess. There are no leading / trailing spaces.

OUTPUT

Output N lines, corresponding to the guesses in the input in order.

For each guess, if it is reasonable, output a standard form of it. Otherwise, output `No :(`.

If a guess has several standard forms, output any.

SAMPLE TESTS

	Input	Output
1	HKOI 2 hOnG KONG olimpiada en informatica HKOI	Hong Kong Olimpiada en Informatica No :(

SUBTASKS

For all cases:

Let L denote the length of acronym.

Let G_i denote the length of the i^{th} guess.

Let W_i denote the number of words of the i^{th} guess.

$1 \leq L \leq 100$, $1 \leq N \leq 100$, $1 \leq G_i \leq 5000$

	Points	Constraints
1	15	$L = 1$, $N = 1$, $G_i = 1$
2	16	$L = 1$, $W_i = 1$
3	17	$L = 1$
4	14	$W_i = L$ All guesses are reasonable
5	10	$W_i \leq L$
6	28	No additional constraints

J172 - CARD GAME

Time Limit: 1.000 s / Memory Limit: 256 MB

Dr. Jones, Alice, and Bob are playing a card game. The game is played using a deck of N cards. The cards are numbered from 1 to N. For convenience, we say card i is bigger than card j if $i > j$ and card i is smaller than card j if $i < j$.

Dr. Jones shuffles the cards and places them on a row, forming a sequence. Alice and Bob must act to maximize their scores.

Alice goes from left to right and chooses A cards in order, such that each one is bigger than the previously chosen card. Specifically, if Alice chooses $C_1, C_2, \ldots, C_A$ in order, then $C_1 < C_2 < \cdots < C_A$.

Bob, similarly, goes from left to right and chooses B cards in order, but, unlike Alice, he needs to choose the cards such that each one is smaller then the previously chosen card. Specifically, if Bob chooses $D_1, D_2, \ldots, D_B$ in order, then $D_1 > D_2 > \cdots > D_B$.

Note that Alice and Bob will not remove the chosen cards from the row.

A player's score is defined as the number of cards he/she has chosen. Alice and Bob are both clever, so they will aim for the highest score.

Although Alice and Bob are having fun, Dr. Jones gets bored pretty soon, having nothing to do other than to shuffle the cards and to watch them play. Therefore, Dr. Jones designs a game for himself.

Suppose Alice's maximal achievable score is A_{max} and Bob's maximal achievable score is B_{max}. Dr. Jones fixes a parameter, K, and his aim is to arrange the cards so that $\max(A_{max}, B_{max}) = K$. In other words, the maximum of Alice's score and Bob's score should equal K.

Dr. Jones figures out how to play this game effortlessly. Can you do the same?

INPUT

The first and only line of input consists of two integers N and K.

OUTPUT

Output a sequence of N cards satisfying $\max(A_{max}, B_{max}) = K$. If there are several solutions, output any.

If there is no solution, output `Impossible` instead.

SAMPLE TESTS

	Input	Output
1	`7 5`	`1 7 6 5 4 2 3`
2	`7 4`	`1 3 5 7 6 4 2`
3	`10 9`	`1 2 3 4 5 6 7 8 10 9`
4	`5 2`	`Impossible`

SUBTASKS

For all cases: $1 \leq K \leq N \leq 100000$

	Points	Constraints
1	17	$1 \leq N \leq 5$
2	11	$1 \leq N \leq 7$
3	9	$1 \leq N \leq 9$
4	20	$1 \leq N \leq 4000$
5	14	$K > \frac{N}{2}$
6	29	No additional constraints

J173 - FIBONACCI WORD

Time Limit: 1.000 s / Memory Limit: 256 MB

It is well known that the Fibonacci sequence is generated using the rules $f(1) = f(2) = 1$ and $f(n+2) = f(n+1) + f(n)$.

In this problem, we consider the so-called Fibonacci word. Denote the i^{th} Fibonacci word by $F(i)$. Then:

1. $F(1)$ = `0`
2. $F(2)$ = `01`
3. $F(n+2)$ = $F(n+1)F(n)$ (concatenation)

For example, $F(3)$ = $F(2)F(1)$ = `010`, $F(4)$ = $F(3)F(2)$ = `01001`.

Note that, for all positive integers m, $F(m)$ is a prefix of $F(m+1)$. Therefore, we may repeat this process of string concatenation to obtain an infinitely long string, F = `0100101001001010010...`. F has the property that, for all positive integers m, $F(m)$ is a prefix of F.

We use $F[l..r]$ to denote the substring consisting of the l^{th} to r^{th} characters of F. For example, $F[1..1]$ = `0`, $F[5..10]$ = `101001`.

Your task is to answer Q queries of the form (L_i, R_i, W_i): in the string $F[L_i..R_i]$, how many occurrences of W_i are there? Here, W_i is one of the followings: `00`, `01`, `10`, and `11`. Occurrences are allowed to overlap, i.e., there are 2 occurrences of `00` in `000`.

INPUT

The first line of input consists of an integer Q.

In the next Q lines, the i^{th} line consists of two integers L_i and R_i, and a binary string W_i, representing the parameters of the i^{th} query.

OUTPUT

Output Q lines.

On the i^{th} line, output the number of occurrences of W_i in the string $F[L_i..R_i]$.

SAMPLE TESTS

	Input	Output
1	4 3 10000 11 1 2 01 1 5 01 2 8 10	0 1 2 3

SUBTASKS

For all cases: $1 \leq Q \leq 10000$, $1 \leq L_i \leq R_i \leq 10^{18}$

	Points	Constraints
1	3	$W_i = \boxed{11}$
2	24	$R_i \leq 2000$
3	17	$R_i \leq 1000000$
4	19	$R_i - L_i \leq 100$
5	37	No additional constraints

J174 - CROSSES

Time Limit: 1.000 s / Memory Limit: 256 MB

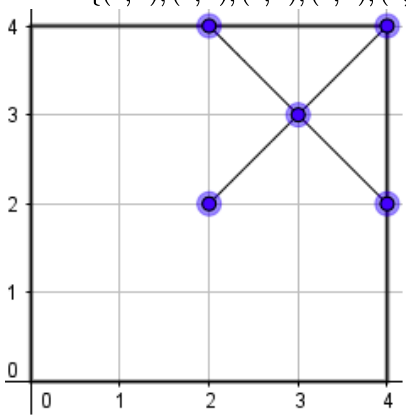
Consider a rectangular grid on the Cartesian coordinate plane with opposite vertices $(0, 0)$ and (N, M) and sides parallel to the grid lines.

Define a cross as a set of points, which can be obtained as follows:

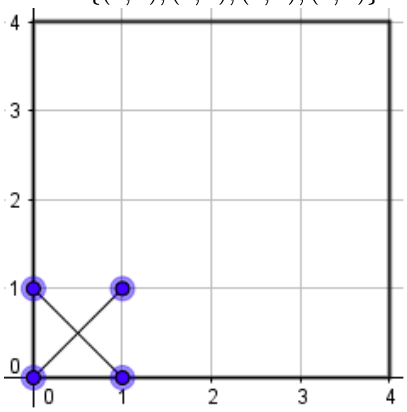
1. Choose an axis-parallel square with positive area, which lies within the grid and whose vertices have integral coordinates.
2. Take all points which lies on at least one of the two main diagonals. A main diagonal is a line segment which connects a pair of opposite vertices of the square.

For example, if $N = M = 4$:

The set $\{(2, 2), (4, 2), (3, 3), (2, 4), (4, 4)\}$ is a cross.



The set $\{(0, 0), (0, 1), (1, 0), (1, 1)\}$ is also a cross.



The sets $\{(1, 0)\}$ and $\{(1, 1), (2, 2), (4, 2), (3, 3), (2, 4), (4, 4)\}$ are not crosses.

Dr. Jones has recently discovered that Alice and Bob are using crosses to communicate secrets. He has also found what he believes to be the "remains" of a cross, which is basically a set of K points, left by them.

To learn of their secrets, Dr. Jones must first figure out what the original cross is. Since blind guessing may be time-consuming, Dr. Jones wants to find out the number of possible crosses left by Alice and Bob. If the number of possibilities is too large, he will have to give up!

Formally, given N, M, and a set of K points on the plane, determine how many different crosses there are, such that each cross contains all the given points.

Two crosses are different if and only if their respective sets of points are different.

Note that it is entirely possible that Alice and Bob just wanted to trick Dr. Jones and left a set of points which is not contained in any cross.

INPUT

The first line of input consists of three integers N, M, and K.

The next K lines of input each consists of two integers x_i, y_i, meaning that the i^{th} point has coordinates (x_i, y_i). It is guaranteed that the K points are pairwise distinct.

OUTPUT

Output one single integer, the number of different crosses which contains the K given points.

SAMPLE TESTS

	Input	Output
1	2 2 1 0 2	2
2	2 3 2 0 0 0 2	1

SUBTASKS

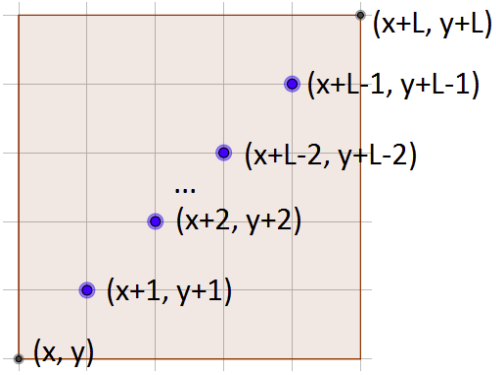
For all cases: $1 \le N, M \le 10^6$, $0 \le K \le 200000$, $0 \le x_i \le N$, $0 \le y_i \le M$

	Points	Constraints
1	18	$1 \le N, M \le 10$
2	12	$1 \le N, M \le 120$
3	13	$1 \le N, M \le 400$
4	15	$1 \le N, M \le 3000$
5	9	$K = 0$
6	11	$K = 1$
7	22	No additional constraints

HINTS

For an axis-parallel square with length L and bottom-left corner located at (x, y):

The coordinates of the points lying on the diagonal connecting the bottom-left and the top-right corners are $(x, y), (x+1, y+1), \ldots, (x+L, y+L)$.



The coordinates of the points lying on the diagonal connecting the top-left and the bottom-right corners are $(x, y+L), (x+1, y+L-1), \ldots, (x+L, y)$.