Problem A. Patrick's Triangle

Source file name:	Triangle.c, Triangle.cpp, Triangle.java, Triangle.py
Input:	`Standard`
Output:	`Standard`

Patrick is an unknown mathematician that wanted to become famous. Patrick had heard that the mathematician Blaise Pascal had became famous because of his triangle, Pascal's triangle.

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

Patrick thought that if he could come up with his own triangle, Patrick's triangle, then he maybe too would become famous. After thinking about it for a long while, he came up with the following construction:

1. Along both the left side and right side of the triangle put the triangle numbers $1, 3, 6, 10, 15, 21, \ldots$

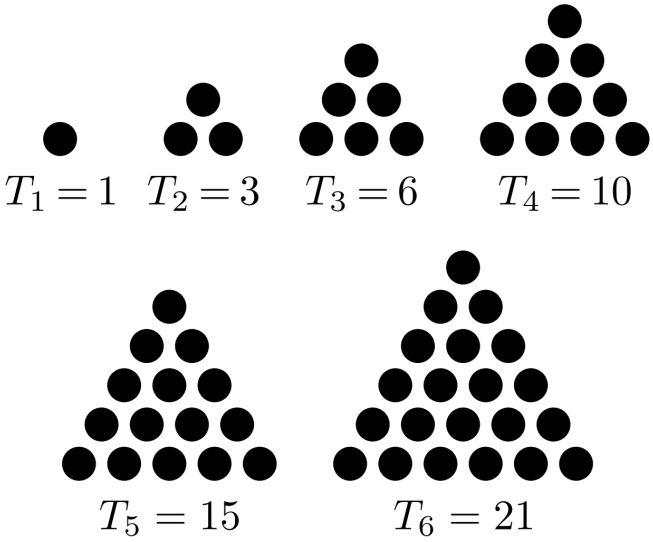
1
3 3
6 6
10 10
15 15

2. Fill in the inside of the triangle by adding the number above and to the left with the number above and to the right, in the same way as Pascal's triangle is constructed.

1
3 3
6 6 6
10 12 12 10
15 22 24 22 15

Patrick knew that in order for his triangle to make him famous, he needed to make sure no one else had invented it before him. So he searched through the On-Line Encyclopedia of Integer Sequences (OEIS.org), and luckily his triangle was missing from OEIS.

Now there was just one thing left to do, to upload his triangle to OEIS and become famous. To do this, Patrick calculated the first 10^6 rows of the triangle by hand. But before he uploaded his triangle to OEIS, he wanted to make sure there were no errors in his calculations. Can you help him check his calculations?



The first 6 triangle numbers. Wikimedia Commons, CC BY-SA 3.0.

Input

The first line consists of one integer Q ($1 \leq Q \leq 2 \cdot 10^5$), the number of queries. The next Q lines consist of three space separated integers N, K, X ($1 \leq K \leq N \leq 10^6$, $0 \leq X < 10^9 + 7$), where X is the value Patrick got when he calculated the K-th number on the N-th row in Patrick's triangle mod $10^9 + 7$. Patrick wants to know if X is correct or not.

Output

For each of the Q queries, output a single line with the string `Correct` if the K-th number on the N-th row in Patrick's triangle is equal to X mod $10^9 + 7$, otherwise output the string `Incorrect`.

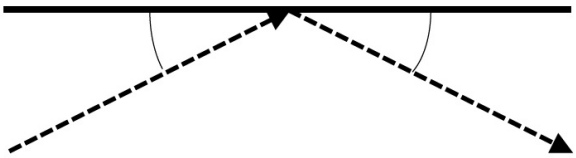
Example

Input	Output
5 4 1 10 4 4 15 3 2 6 6 2 37 123456 61728 664742090	Correct Incorrect Correct Correct Correct

Problem B. Square Bounce

Source file name:	Bounce.c, Bounce.cpp, Bounce.java, Bounce.py
Input:	`Standard`
Output:	`Standard`

Given a square in the plane with corners at $(-1,\ -1)$, $(-1,\ 1)$, $(1,\ 1)$ and $(1,\ -1)$, we fire a ray from point $(-1,\ 1)$ into the interior of the square on a path with a given slope. The ray bounces off of the sides of the square with an angle of reflection which is the same as the angle of incidence to the side at the point of intersection.



After a number of bounces, the ray intersects one of the square's sides again at some point which has rational coordinates. Find those rational coordinates in reduced form.

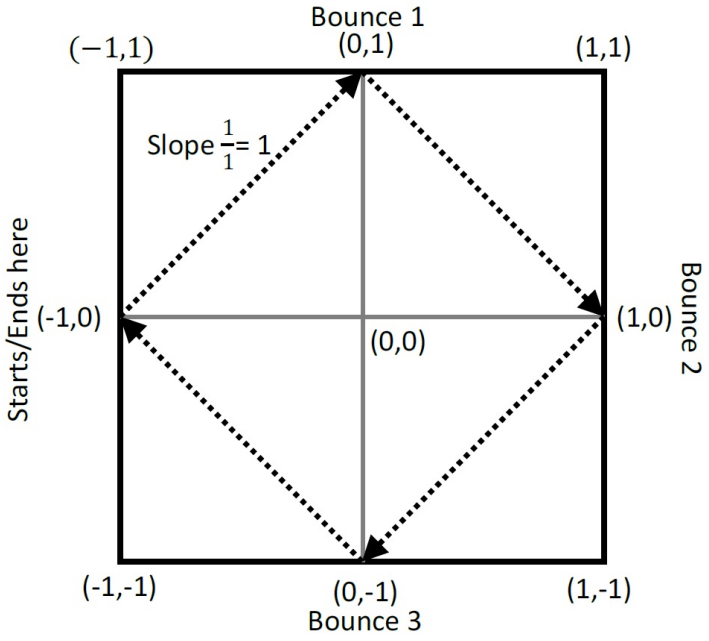
Input

The single line of input contains three integers a, b and n ($1 \leq a,\ b,\ n \leq 10^6,\ gcd(a,\ b) = 1$), where the slope of the ray's initial path is a/b, and there are n bounces. Note that a and b are relatively prime. The slope will be chosen so that the ray never bounces at a corner of the square.

Output

Output a single line with four space-separated integers p, q, s and t, where $(p/q,\ s/t)$ is the final point where the ray hits a side of the square, p/q and s/t are in reduced form, and the denominators (q and t) are positive. If one of the coordinates has value 0, output it as 0 1.

The following is a picture of the first example:

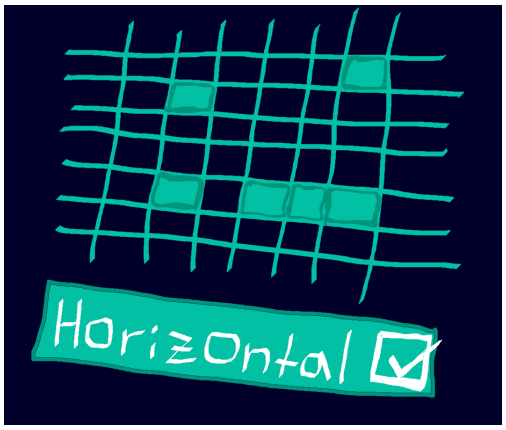


Example

Input	Output
1 1 3	-1 1 0 1
1 7 4	-1 1 6 7
355 113 123456	1 1 -58 113

Problem C. Bar Classification

Source file name:	Barclassification.c, Barclassification.cpp, Barclassification.java, Barclassification.py
Input:	`Standard`
Output:	`Standard`



You are taking a course on machine learning at your university, and as homework you have been tasked with writing a program that can tell vertical bars from horizontal bars in images. To generate some training data, you use the following method. First, take an $N \times N$ grid, and fill it with zeros. Next, take a row or a column, and fill it with ones. Finally, take at most N arbitrary cells, and flip them. Flipping a cell means changing a zero to a one, or changing a one to a zero.

Generating data this way is easy, but how to generate all the answers? It will take hours to go through the training data manually. If only you had a program that finds all the outputs automatically somehow.

You are given an $N \times N$ matrix that has been generated as in the description. Write a program that finds whether it was a column or a row that was filled with ones, or if it is impossible to determine.

Input

The first line of input consists of an integer N ($2 \leq N \leq 1000$), the size of the grid.

The following N lines each contain a string of length N consisting of zeros and ones. These are the rows of the grid.

It is guaranteed that the input was generated by taking a grid of zeros, putting ones on a row or a column, and then flipping at most N cells.

Output

If the bar was vertical (a column), print "|". If it was horizontal (a row), print "-". If it is impossible to determine (because it could be both), print "+".

Example

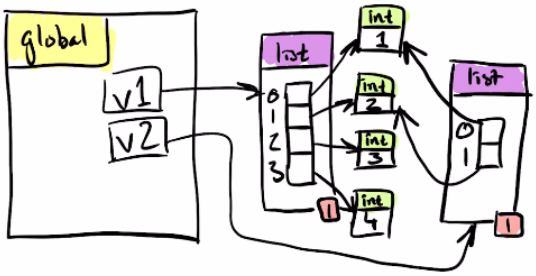
Input	Output
5 01100 01000 01001 00000 01000	|
3 111 000 111	-
3 010 101 010	+

Problem D. Box and Arrow Diagram

Source file name:	Diagram.c, Diagram.cpp, Diagram.java, Diagram.py
Input:	`Standard`
Output:	`Standard`

What an embarrassment! Itaf got 0/5 points in her last "Fundamental programming in Python" exam. She studies Engineering physics at KTH and is struggling with this course. She is not alone, as 60% of her classmates failed the exam this year. The reason for this oddly high percentage is the so called *box and arrow diagram* (låd- och pildiagram).

In this part of the exam you are given a piece of Python code and you have to draw how the memory structure will look like when the program reaches a given line. Since Itaf is a high-rated competitive programmer her ego always came in the way whenever she tried to study for the test, because it felt "too easy". But now she has become desperate and needs your help.



An example of a box and arrow diagram, taken from github.com/dicander/box_arrow_diagram

The *box and arrow diagram* is used to explain the memory structure inside Python. Simplified, the diagram can be seen as a directed graph with nodes (boxes) labeled from 1 to N and edges (arrows) labeled from 1 to M. The boxes corresponds to the objects in the memory of a Python program. Box 1 is special, it represents the *global* object. An arrow being drawn from box u to box v in the diagram means that object u stores a reference of object v. If u stores multiple references of v, then you draw multiple arrows from u to v. It is also possible for an object to contain references to itself.

An object u is said to be *alive* if there exists a path from the *global* object to u in the *box and arrow diagram*. Each object also has a reference counter. The reference counter of an object u is defined as the number of arrows (v, u) such that v is alive.

Itaf now needs your help, and she will ask you Q queries, each query can be one of two types.

- `1 X` Remove the arrow with label X from the diagram.
- `2 Y` Output the reference counter of the object with label Y.

Input

The first line consists of two space separated integers N, M $(1 \leq N, M \leq 2 \cdot 10^5)$, where N is the number of boxes in the diagram and M is the number of arrows in the diagram.

The next M lines describe the arrows in the diagram. The i-th line contains 2 space separated integers U_i, V_i $(1 \leq U_i, V_i \leq N)$, meaning the arrow with label i goes from box U_i to box V_i. Note that arrows forming loops and multi-edges are allowed.

The next line contains an integer Q $(1 \leq Q \leq 2 \cdot 10^5)$, the number of queries. The next Q lines describe the Q queries. The j-th query is given as a pair of space separated integers C_j, X_j $(1 \leq C_j \leq 2)$.

- If $C_j = 1$ then remove the arrow labeled X_j from the diagram $(1 \leq X_j \leq M)$.
- If $C_j = 2$ then output the reference counter of object X_j $(1 \leq X_j \leq N)$.

It is guaranteed that there will not be two queries of type 1 with same value of X_j, meaning the same arrow will never be deleted twice.

Output

For each query of type 2, output a single line containing the reference count of object Y_j.

Example

Input	Output
3 4 1 2 2 3 1 2 3 3 7 2 2 2 3 1 4 2 3 1 1 1 3 2 3	2 2 1 0

Problem E. 99 Problems

Source file name:	Problems99.c, Problems99.cpp, Problems99.java, Problems99.py
Input:	`Standard`
Output:	`Standard`

Ingrid is the founder of a company that sells bicycle parts. She used to set the prices of products quite arbitrarily, but now she has decided that it would be more profitable if the prices end in 99.

You are given a positive integer N, the price of a product. Your task is to find the nearest positive integer to N which ends in 99. If there are two such numbers that are equally close, find the bigger one.

Input

The input contains one integer N ($1 \leq N \leq 10^4$), the price of a product. It is guaranteed that the number N does not end in 99.

Output

Print one integer, the closest positive integer that ends in 99. In case of a tie, print the bigger one.

Example

Input	Output
10	99
249	299
10000	9999

Problem F. Shortest Missing Subsequences

Source file name:	Shortest.c, Shortest.cpp, Shortest.java, Shortest.py
Input:	`Standard`
Output:	`Standard`

Given a string s we say that string t is a *Subsequence* of s if t can be obtained from s by deleting zero or more characters of s. Note that t is not necessarily a substring of s–that is, t is not necessarily contiguous in s, but the characters of t appear in the same order as they do in s.

For a given subset, v, of the lowercase English alphabet characters from 'a' to 'z', we say that string u is a *Missing Subsequence* of another string s if u is not a *Subsequence* of s, but all characters in u and all the characters of s are in the set v. A *Shortest Missing Subsequence* of s is a *Missing Subsequence* of s with the smallest length among all *Missing Subsequences* of s.

Given a set of English alphabetic characters, a target string made up of characters from that set, and a list of query strings made up of characters from that set, determine if each of the query strings is a *Shortest Missing Subsequence* of the target string.

Input

The first line of input contains a string v ($1 \leq |v| \leq 26$) of lowercase letters, in lexicographical order. Each letter appears at most once. This is the set of alphabetic characters.

The next line of input contains a string s ($1 \leq |s| \leq 10^6$, s only contains letters from v). This is the target string to be queried.

The next line contains an integer n ($1 \leq n \leq 10^6$). This is the number of queries.

Each of the next n lines contains a string q ($1 \leq |q| \leq 10^6$, q only contains letters from v). These are the query strings. The sum of the lengths of all query strings will not exceed 10^6.

Output

Output n lines, one for each query. On each line, output either 1 if the query string is a *Shortest Missing Subsequence* of the target string, or 0 if it is not. The outputs must be in the order of the input queries.

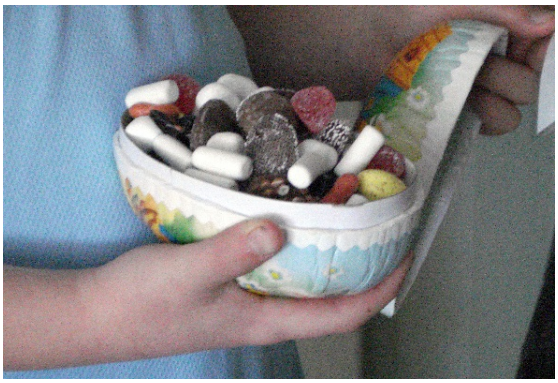
Example

Input	Output
abc abcccabac 3 cbb cbba cba	1 0 0

Problem G. The Great Egg Hunt

Source file name:	Greategg.c, Greategg.cpp, Greategg.java, Greategg.py
Input:	`Standard`
Output:	`Standard`

Every year at Easter Bob's grandmother organizes *The Great Egg Hunt* in their family mansion. This has been a family tradition since even before Bob was born. Bob's grandmother starts by filling up a giant Easter Egg with candy. She then picks a room (uniformly) at random and hides the egg there. First person to find the egg gets to keep all the candy inside.



Easter Egg with Candy. Wikimedia Commons, CC BY-SA 3.0.

Bob's family mansion consists of N rooms. There are $N-1$ doorways between pairs of rooms. You can assume that the mansion is connected, meaning it is always possible to walk from any room to any other room using the doorways.

As Bob is a veteran egg hunter, Bob has developed a method for finding the egg, which he calls *Egg First Search*:

1. If Bob is currently in an unexplored room, then he searches that room. Since Bob is a veteran egg hunter, if the egg is in that room, then he will find it.
2. Otherwise, Bob moves to an adjacent room that is in the direction of any of the closest unexplored rooms. If there are multiple rooms he could move to, then he picks one (uniformly) at random.

Assume it takes 1 unit of time for Bob to search a room, and also assume that it takes 1 unit of time for Bob to move from a room to an adjacent room.

The one thing Bob is still not sure about is in which room he should start his search. So he asks you for help. Given a map of the mansion, find which starting room(s) that minimize the expected time to find the egg using Bob's Egg First Search method.

Input

The first line consists of one integer N ($1 \leq N \leq 2 \cdot 10^5$), the number of rooms in the mansion. The next $N-1$ lines consist of two space separated integers u and v ($1 \leq u, v \leq N$, $u \neq v$), a pair of rooms that are connected by a doorway. It is guaranteed that the mansion is connected, meaning it is always possible to walk from any room to any other room using the doorways.

Output

On the first line print an integer M, the number of optimal starting rooms. On the second line, print M space separated integers $S_1, \ldots, S_M$ ($1 \leq S_1 < S_2 < \ldots < S_M \leq N$), the list of optimal starting rooms.

Example

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

Problem H. Tree Hopping

Source file name:	Hopping.c, Hopping.cpp, Hopping.java, Hopping.py
Input:	`Standard`
Output:	`Standard`

You are given a tree and a permutation of its vertices. It can be proven that for any tree and any pair of source/destination nodes, there is some permutation of the nodes where the first node is the source, the last node is the destination, and the distance between adjacent nodes in the permutation is less than or equal to three.

Your job will be to write a verifier for this property. Given such a permutation and the tree, validate whether the distance between adjacent nodes in the permutation is less than or equal to three.

Input

The first line of input contains an integer t ($1 \leq t \leq 5 \cdot 10^4$), which is the number of test cases.

In each test case, the first line of input contains an integer n ($2 \leq n \leq 10^5$), which is the number of nodes in the tree. The nodes are numbered from 1 to n.

Each of the next $n-1$ lines contains a pair of integers a and b ($1 \leq a < b \leq n$), representing an edge in the tree between nodes a and b.

Each of the next n lines contains an integer p ($1 \leq p \leq n$, all values distinct). This is the permutation of the nodes.

The sum of the values of n over all test cases will not exceed 10^5.

Output

For each test case, output a single line with a single integer, which is 1 if the given permutation satisfies the constraint that every pair of adjacent nodes in the permutation has distance less than or equal to three in the tree. Output 0 if the given permutation does not satisfy this constraint.

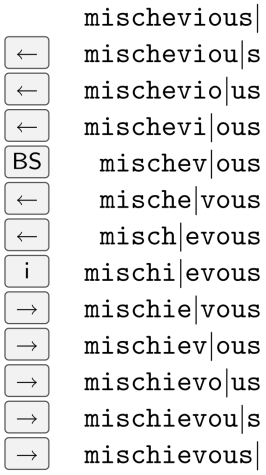
Example

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

Problem I. Sperhling

Source file name:	Sperhling.c, Sperhling.cpp, Sperhling.java, Sperhling.py
Input:	`Standard`
Output:	`Standard`

On her spare time Caitlin loves to do speed typing. Unfortunately Caitlyn was never good at spelling. So what often happens is that Catelin makes a mistake spelling a word, and has to go back and fix her typo. One word in particular that Caitlynn has a hard time spelling is the word `mischievous`. Catelyn usually spells it `mischevious` instead. In order to fix this typo, Caytlin needs to do 12 key presses.



Katelin thinks that no matter how much time she puts into speed typing, she will always make mistakes. But maybe she can get better at quickly fixing her mistakes. Katelyn would like you to help her with this.

Suppose that Caytlyn has just written the word S_1, but that she should have written S_2. Assume that the cursor is initially at the end of S_1, and that after fixing the typo, Caitlin needs to place the cursor at the end of S_2. Output the fewest number of key presses needed to make S_1 into S_2.

The keys you are allowed to use are

- the left and right arrow keys.
- the back space key.
- the keys $\texttt{a}, \ldots, \texttt{z}$.

You are not allowed to press back space or the left arrow key when the cursor is all the way to the left. You are also not allowed to press the right arrow key when the cursor is all the way to the right.

Input

The first two lines contain the strings S_1 respectively S_2. Both S_1 and S_2 consist only of lowercase English letters, and are between 1 and 100 characters long.

Output

Output a single line with the fewest number of key presses needed to make S_1 into S_2.

Example

Input	Output
mischevious mischievous	12
cerstermergerd customer	20
caitlin caitlynn	5

Problem J. Tournament Seeding

Source file name:	Tournament.c, Tournament.cpp, Tournament.java, Tournament.py
Input:	`Standard`
Output:	`Standard`

You are tasked with seeding a single-elimination tournament for a one-on-one game. The number of players who have registered for the tournament is exactly a power of two, and there will be exactly enough rounds in this tournament to decide a winner. Furthermore, each player has a unique numeric rating in the game known to you; when two players play against each other in a game, the player with the higher rating always wins. As the organizer of the tournament, you would like to make the tournament as exciting for players and spectators as possible. To do that, you wish the tournament to have the following properties:

- The top two (highest rated) players are present in the final round of the tournament, the top four players are present in the semi-final round of the tournament, the top eight players are present in the quarter-final round, and so on. This saves the highest rated games for last.
- Subject to the above, as many games as possible are "close." We define a game to be "close" if the difference between the two players' ratings is less than or equal to some threshold.

Given the number of rounds, the threshold for "close" games and the ratings of the players, what is the maximum number of "close" games that can happen subject to the above constraints?

Input

The first line of input contains two integers n ($1 \leq n \leq 18$) and k ($1 \leq k \leq 10^9$), where n is the number of rounds of the tournament, and k is the rating difference that makes a game "close."

Each of the next 2^n lines contains a single integer r ($1 \leq r \leq 10^9$) denoting the rating of each player. The ratings are guaranteed to be distinct.

Output

Output a single line with a single integer, which is the maximum number of "close" games possible in a tournament among these players satisfying the constraints described above.

Example

Input	Output
2 2 9 1 6 4	1
2 5 9 1 6 4	3