

Problem A. AHB

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

A and B are two non-negative integer numbers that both consist of n digits. Let the i -th digit of a number X be X_i . The operation $A \text{ } || \text{ } B$ gives an integer number defined as follows: the i -th digit of $A \text{ } || \text{ } B$ is equal to $|A_i - B_i|$. Calculate the result of $A \text{ } || \text{ } B$!

Input

The only line of the input contains two space-separated integers A, B ($0 \leq A, B \leq 10^9$). It is guaranteed that A and B have an equal number of digits.

Output

Output a single integer — the result of $A \text{ } || \text{ } B$, without leading zeroes.

Examples

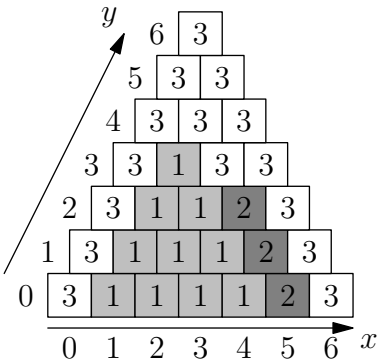
standard input	standard output
123 151	32
11 21	10

Problem B. Wet Boxes

Input file: **standard input**
Output file: **standard output**
Time limit: 0.7 seconds
Memory limit: 256 megabytes

Bob works in a warehouse which contains a large pile of boxes. The position of a box can be described with a pair of integers (x, y) . Each box either stands on the ground ($y = 0$) or stands on top of two boxes with positions $(x, y - 1)$ and $(x + 1, y - 1)$ (see the figure).

Sometimes the contents of a box leak out and the box gets wet. When a box becomes wet, so do the two boxes below it. Given a list of boxes that leak in succession, help Bob count how many dry boxes became wet after each leak. Don't include boxes that were already wet.



Input

The first line contains a single integer n , the number of leaking boxes ($1 \leq n \leq 10^5$). The i -th of the next n lines contains two space-separated integers x_i and y_i , the position of the i -th leaking box ($0 \leq x_i, y_i \leq 10^9$).

Output

Output n lines: in the i -th line output the number of boxes that became wet after the i -th box had leaked.

Examples

standard input	standard output
4	10
1 3	3
3 2	15
0 6	0
1 1	

Problem C. Minimax Tree

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Bob’s new favourite toy is a rooted tree that consists of n vertices numbered from 1 to n . The number of the root vertex is 1. The tree has l leafs (the root is not considered to be a leaf). Each leaf of the tree has an integer written in it.

This birthday Bob received $n - l$ stickers as a gift: k of them are labelled “min”, and the other $n - l - k$ are labelled “max”. Bob has decided to place the stickers on the internal vertices of the tree, a single sticker on each internal vertex.

Once he has placed all the stickers on the tree, Bob would like to calculate a function f for each vertex v of the tree in the following fashion:

- If v is a leaf, $f(v)$ is equal to the integer that is written in v .
- If v has a “min” sticker, $f(v)$ is equal to the minimum value of $f(u)$, where u is any child of v .
- If v has a “max” sticker, $f(v)$ is equal to the maximum value of $f(u)$, where u is any child of v .

Bob isn’t yet sure how to place his stickers on the tree, but he is interested in the value of f in the root vertex. Given the tree and the stickers, help Bob calculate the minimum and the maximum possible value of $f(1)$!

Input

The first line contains two space-separated integers n and k ($2 \leq n \leq 10^5$, $0 \leq k \leq n$). The second line contains $n - 1$ space-separated integer numbers p_2, p_3, \dots, p_n ($1 \leq p_i \leq n$). The number p_i denotes the parent of the vertex numbered i . The third line contains n space-separated integer numbers a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$). If the vertex i is a leaf, then a_i is the number written in that vertex. Otherwise a_i will be equal to 0.

It is guaranteed that the given graph will be a tree. It is guaranteed that $k + l \leq n$.

Output

In a single line output two integers separated by a space — the minimum and the maximum possible value of $f(1)$.

Examples

standard input	standard output
6 1 1 1 2 2 3 0 0 0 1 3 2	2 3

Note

A tree is a connected graph that has no cycles. A rooted tree is a tree with one vertex being the root vertex. In a rooted tree, a vertex u is a child of v if and only if there is an edge between v and u , and u does not belong to the path that connects the root vertex with v . The vertex v then is called the parent of u . A vertex of a rooted tree is called a leaf if and only if it has no children. Otherwise the vertex is called an internal vertex.

Problem D. Journey

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **256 megabytes**

Alice is playing a popular computer RPG game “Grid Fantasy”. The world map in this game is a rectangle with n columns and m rows. Each cell in the map can be either a town, wilderness or mountains. Furthermore, any non-mountain cell that is border-adjacent to a town is wilderness, and any non-mountain cell that is border-adjacent to wilderness is a town. Players can move between border-adjacent cells, but cannot enter mountain cells. Moving from wilderness to a town costs a gold, and moving from a town to wilderness costs b gold.

Initially Alice is in the top left cell of the map, which is a town. Alice wants to move to the bottom right cell of the map. Help Alice find the minimum amount of gold she needs to do this.

Input

The first line of the input contains two integers n and m ($1 \leq n, m \leq 500$) — the width and the height of the map. The second line contains two integers a and b ($0 \leq a, b \leq 1000$) — the cost of entering a town and the cost of entering wilderness.

The next m lines each contain a string of n characters. The i -th character in the j -th string contains “.” if the corresponding cell can be occupied by a player and “#” if it is a mountain cell.

Output

The output should contain a single integer — the minimum amount of gold needed to move to the bottom right cell. If it is impossible, print “IMPOSSIBLE”.

Examples

standard input	standard output
5 3 1 2 .#... .#.#. ...#.	15
3 3 4 5 ..# .#. #..	IMPOSSIBLE

Problem E. Permutation Polygon

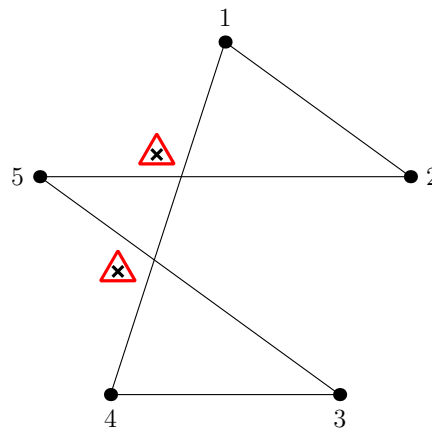
Input file: standard input
Output file: standard output
Time limit: 0.5 seconds
Memory limit: 256 megabytes

There are n cities in Regularia, Alice's homeland. They are located at the vertices of a regular n -gon. The cities are numbered from 1 to n in a clockwise order (see the figure).

The people of Regularia love Alice so much that they made her the queen of the country. Alice wants to prove herself, so she has ordered a construction of a new highway system connecting cities of Regularia. Alice has come up with a plan to build the highways, which is an n -permutation p_1, p_2, \dots, p_n . Specifically, for a city numbered i , Alice wants to build a highway between the cities i and p_i , where p_i is the number of some other city. A highway is a straight line connecting two cities. According to Alice's plan, any two cities will be connected by no more than a single highway.

Of course, some of the highways may intersect one another. Two highways are said to intersect if and only if there is a unique point that belongs to both of the highways and is not a city. This point is called a junction of the two highways. Even if more than two highways meet at a single point, each pair has its own separate junction. As a complicated and expensive interchange has to be built at each junction, Alice is interested in the total number of junctions in her plan. Help her find this number!

For example, the plan for a permutation 4, 1, 5, 3, 2 is shown in the figure.



Input

The first line contains a single integer n , the number of cities in Regularia ($3 \leq n \leq 10^5$). The next line contains n space-separated integers p_1, p_2, \dots, p_n , which is a permutation of length n . It is guaranteed that $i \neq p_i$ for any i and that no two cities will be connected by more than a single highway.

Output

Output a single integer — the total number of junctions in Alice's plan.

Examples

standard input	standard output
5 4 1 5 3 2	2
6 4 5 6 3 1 2	6

Note

A permutation of length n is any rearrangement of the sequence $1, 2, \dots, n$.

Problem F. Unusual Sum

Input file: `standard input`
Output file: `standard output`
Time limit: 1.5 seconds
Memory limit: 256 megabytes

Bob is an aspiring archaeologist. He has found n ancient scrolls, each with an unusual sum written on it. The sum on the i -th scroll reads as

$$\sum_{k=l_i}^{r_i} \frac{1}{k^2 + k}.$$

Bob thinks that precise values of these sums encode the location of Aldoredó, the not-so-famous city of gold. Help him calculate the values of the sums!

Input

The first line contains a single integer n , the number of scrolls ($1 \leq n \leq 10^5$). The i -th of the next n lines contains two space-separated integer numbers l_i and r_i , the description of the sum on the i -th scroll ($1 \leq l_i \leq r_i \leq 10^{18}$).

Output

Output n lines: in the i -th line output the value of the sum written on the i -th scroll. Your answer will be considered correct if its relative or absolute error doesn't exceed 10^{-9} .

Examples

standard input	standard output
3	0.666666666667
1 2	0.016424751719
34 76	0.050000000000
4 4	

Problem G. Robot Walk

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Bob has built a tiny robot whose movement he can program. He also has a board that consists of n cells lined up in a row, numbered 1 to n from left to right. Each of the cells also has a single letter written on it. Initially Bob places the robot in the cell number x . Then Bob executes a program that moves the robot on the board. One instruction of the program will move the robot either one cell to the left or one cell to the right.

The robot also has a photo sensor directed at the board, and whenever Bob changes the location of the robot (including placing it in the initial cell), it recognizes and sends the letter that is written in the cell to Bob's computer. After the robot finishes the program, the computer prints the received text.

Given the program of m moves for the robot, print out the text the computer will output. The program consists of characters "L" and "R", which mean that the robot should move one cell to the left or to the right, respectively.

Input

The first line contains two space-separated integers n and x ($2 \leq n < 10^5, 1 \leq x \leq n$). The second line contains a string s of length n : the i -th character of s is the letter written in the cell number i . The string s will consist of lowercase latin letters. The third line contains a single integer m — the length of the program ($1 \leq m \leq 10^5$). The fourth line contains the program consisting of symbols "L" and "R".

It is guaranteed that the program will not cause the robot to leave the board.

Output

In a single line output the text printed out by the computer.

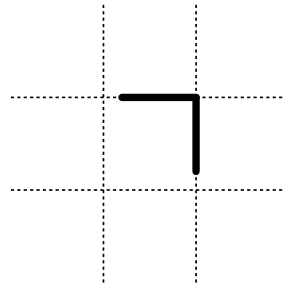
Examples

standard input	standard output
7 1 taconut 10 RRRLRLRRRR	tacococonut

Problem H. Game of Corners

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Alice is playing the Game of Corners. She is given a rectangular grid with n rows and m columns. During one move, Alice can choose any two previously unused cell borders that have a common point and join them in a corner (pictured). A corner can be oriented in any way.



Initially the grid is clear of corners. How many moves can Alice possibly make?

Input

The only line of the input contains two integers n and m ($1 \leq n, m \leq 10^9$) — the number of rows and the number of columns in the grid.

Output

Output a single integer — the maximum number of moves Alice can make.

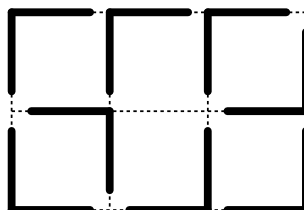
Examples

standard input	standard output
2 3	8
1 1	2

Note

Two corners can share a point on the grid, but cannot share a cell border.

For the first given sample, one possible arrangement of corners is:



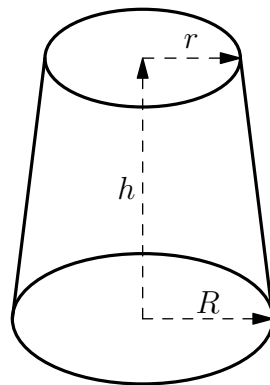
Problem I. Shell Game

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Bob has discovered a new quality in himself — he likes to trick people a lot! In particular, Bob wants to try his tricking skills in the shell game.

The shell game involves three identical cups and a single ball. In the beginning the host (that's Bob!) puts the cups upside down and places the ball inside one of the cups. Then he shuffles the cups and the player has to guess where the ball is. The host then lifts that cup, revealing whether the ball was inside or not. The player wins if he makes a correct guess, otherwise the host wins.

The form of each cup is a sliced cone. Formally, the base of the cup is a circle with radius r , and the opening of the cup is a circle with radius R . The height of the cup is equal to h . A ball for the shell game is simply a sphere.



What the player is not going to know is that Bob smeared the inner surface of each cup with glue, resulting in the cup holding the ball when Bob lifts it up — so the player will never win! However, the ball will stick only if it touches the inner surface of a cup. For this reason Bob wants to get the largest possible ball that fits inside the cup when the cup is placed upside down on the table.

Bob has already found three identical cups in his grandmother's locker — now he only has to buy a ball of the required size so that he may start playing the shell game. Help Bob and calculate the largest size of such a ball!

Input

The first line contains three space-separated integers r , R and h , the radii of the base and the opening, and the height of the cup ($1 \leq r < R \leq 10^4$, $1 \leq h \leq 10^4$).

Output

Output a single number — the radius of the largest ball that can fit in the cup. Your answer will be considered correct if its relative or absolute error doesn't exceed 10^{-6} .

Examples

standard input	standard output
3 4 8	3.531128874149

Problem J. Narrow Bus

Input file: **standard input**
Output file: **standard output**
Time limit: 0.5 seconds
Memory limit: 256 megabytes

The narrow bus is so narrow it doesn't have any seats and the passengers have to stand in a row. The bus has two doors that can be used for entering or leaving the bus: one in the front and one in the back. People can't easily swap in the bus because it is too narrow, so the order people are standing in does not change while the bus is running.

At each bus stop, either a single person enters the bus or a single person leaves the bus.

- If a person wants to enter the bus, they will choose one of the doors to do that. Then this person joins the beginning or the end of the row, according to which door they used to enter.
- If a person wants to leave at a stop, they will choose the direction with the least amount of people and proceed to that door (in case both directions have the same amount of people, the person chooses the front door). Everyone in that direction will have to get out then as well. No one likes standing outside and waiting so everybody who wants to continue riding the bus will use the other door to get back in. These passengers will enter the bus back in the same order they got out.

Given a description of people entering and leaving the bus, count how many people will have to get out when somebody leaves the bus.

Input

The first line of input contains a single integer n , the number of actions at bus stops ($1 \leq n \leq 10^5$). The next n lines will contain one of the three types of actions each. The actions happen consecutively as they are given in the input.

- "F" — a person enters the bus using the front door.
- "B" — a person enters the bus using the back door.
- "0 i " — the i -th person that has entered the bus today leaves.

Each action happens at a single bus stop. After the action is completed, the bus leaves for the next stop.

Output

Print a single integer for each type "0" query, the number of people who will have to get out and go back in the bus when a person leaves at the stop.

Examples

standard input	standard output
9	1
F	2
B	
B	
B	
B	
0 2	
F	
F	
0 4	

Note

In the example, after everyone gets in the bus during the first 5 stops, the people are ordered 1, 2, 3, 4, 5.

At the 6th stop, the 2nd person leaves through the front door and the 1st person is forced to get out of the bus. The 1st person then uses the back door to enter the bus again. The order changes to 3, 4, 5, 1.

During the next two stops two people use the front door to enter the bus, and the order becomes 7, 6, 3, 4, 5, 1.

At the 9th stop, the 4th person leaves through the back door and now the 5th and the 1st person are forced to get out of the bus. They then use the front door to get back in the bus. The order changes to 5, 1, 7, 6, 3.

Problem K. Profact

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Alice is bored out of her mind by her math classes. She craves for something much more exciting. That is why she invented a new type of numbers, the profacts. Alice calls a positive integer number a profact if it can be expressed as a product of one or several factorials.

Just today Alice received n bills. She wonders whether the costs on the bills are profact numbers. But the numbers are too large, help Alice check this!

Input

The first line contains a single integer n , the number of bills ($1 \leq n \leq 10^5$). Each of the next n lines contains a single integer a_i , the cost on the i -th bill ($1 \leq a_i \leq 10^{18}$).

Output

Output n lines, on the i -th line output the answer for the number a_i . If the number a_i is a profact, output “YES”, otherwise output “NO”.

Examples

standard input	standard output
7	YES
1	YES
2	NO
3	YES
8	YES
12	YES
24	NO
25	

Note

A factorial is any number that can be expressed as $1 \cdot 2 \cdot 3 \cdot \dots \cdot k$, for some positive integer k , and is denoted by $k!$.

Problem L. Emoticons

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Bob has received a text message from his girlfriend Alice. The message consists only of colons (":") and brackets "(" and ")". Two consecutive characters in the message form an emoticon if one of them is a colon and the other one is a bracket. If the bracket's inner side faces the colon, the emoticon is a smiley; otherwise it is a frowney. So, ":)" and "(:" are smilies, and ":(" and "):" are frownies. One character can belong to two emoticons: for example, there are both a smiley and a frowney in the message "):)".

Unfortunately, Bob has trouble understanding the message. He can only find out Alice's mood based on the number of emoticons in the message:

- If there are more smilies than frownies, Alice is happy.
- If the number of smilies is equal to the number of frownies, Alice is just texting him because she's bored.
- If there are more frownies than smilies, Alice is sad.

Help Bob determine how does Alice feel!

Input

The first line contains a single integer n , the length of the message ($1 \leq n \leq 10^5$). The next line contains the message, a string consisting of n characters. Each of the characters is either ":", "(", or ")"

Output

In a single line, output either "HAPPY", "BORED" or "SAD" corresponding to Alice's mood.

Examples

standard input	standard output
10 :)))))))))	HAPPY
1 :	BORED
8)::(:():	SAD