Codeforces Round #168 (Div. 2)

A. Lights Out

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

Lenny is playing a game on a 3×3 grid of lights. In the beginning of the game all lights are switched on. Pressing any of the lights will toggle it and all side-adjacent lights. The goal of the game is to switch all the lights off. We consider the toggling as follows: if the light was switched on then it will be switched off, if it was switched off then it will be switched on.

Lenny has spent some time playing with the grid and by now he has pressed each light a certain number of times. Given the number of times each light is pressed, you have to print the current state of each light.

Input

The input consists of three rows. Each row contains three integers each between 0 to 100 inclusive. The j-th number in the i-th row is the number of times the j-th light of the i-th row of the grid is pressed.

Output

Print three lines, each containing three characters. The j-th character of the i-th line is "1" if and only if the corresponding light is switched on, otherwise it's "0".

Sample test(s)

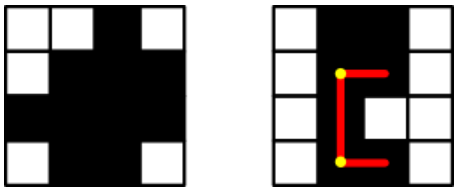
input
1 0 0 0 0 0 0 0 1
output
001 010 100

input
1 0 1 8 8 8 2 0 3
output
010 011 100

B. Convex Shape

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

Consider an $n \times m$ grid. Initially all the cells of the grid are colored white. Lenny has painted some of the cells (at least one) black. We call a painted grid *convex* if one can walk from **any** black cell to **any another** black cell using a path of side-adjacent black cells changing his direction at most once during the path. In the figure below, the left grid is convex while the right one is not convex, because there exist two cells which need more than one time to change direction in their path.



You're given a painted grid in the input. Tell Lenny if the grid is convex or not.

Input

The first line of the input contains two integers n and m $(1 \leq n, m \leq 50)$ — the size of the grid. Each of the next n lines contains m characters "B" or "W". Character "B" denotes a black cell of the grid and "W" denotes a white cell of the grid.

It's guaranteed that the grid has at least one black cell.

Output

On the only line of the output print "YES" if the grid is convex, otherwise print "NO". Do not print quotes.

Sample test(s)

input
3 4 WWBW BWWW WWWB
output
NO

input
3 1 B B W
output
YES

C. k-Multiple Free Set

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

A k-multiple free set is a set of integers where there is no pair of integers where one is equal to another integer multiplied by k. That is, there are no two integers x and y $(x < y)$ from the set, such that $y = x \cdot k$.

You're given a set of n distinct positive integers. Your task is to find the size of it's largest k-multiple free subset.

Input

The first line of the input contains two integers n and k ($1 \le n \le 10^5$, $1 \le k \le 10^9$). The next line contains a list of n distinct positive integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

All the numbers in the lines are separated by single spaces.

Output

On the only line of the output print the size of the largest k-multiple free subset of $\{a_1, a_2, ..., a_n\}$.

Sample test(s)

input
6 2 2 3 6 5 4 10
output
3

Note

In the sample input one of the possible maximum 2-multiple free subsets is {4, 5, 6}.

D. Zero Tree

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

A *tree* is a graph with n vertices and exactly n - 1 edges; this graph should meet the following condition: there exists exactly one shortest (by number of edges) path between any pair of its vertices.

A *subtree* of a tree T is a tree with both vertices and edges as subsets of vertices and edges of T.

You're given a tree with n vertices. Consider its vertices numbered with integers from 1 to n. Additionally an integer is written on every vertex of this tree. Initially the integer written on the i-th vertex is equal to v_i. In one move you can apply the following operation:

1. Select the subtree of the given tree that includes the vertex with number 1.
2. Increase (or decrease) by one all the integers which are written on the vertices of that subtree.

Calculate the minimum number of moves that is required to make all the integers written on the vertices of the given tree equal to zero.

Input

The first line of the input contains n ($1 \le n \le 10^5$). Each of the next n - 1 lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$; $a_i \ne b_i$) indicating there's an edge between vertices a_i and b_i. It's guaranteed that the input graph is a tree.

The last line of the input contains a list of n space-separated integers v_1, v_2, ..., v_n ($|v_i| \le 10^9$).

Output

Print the minimum number of operations needed to solve the task.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

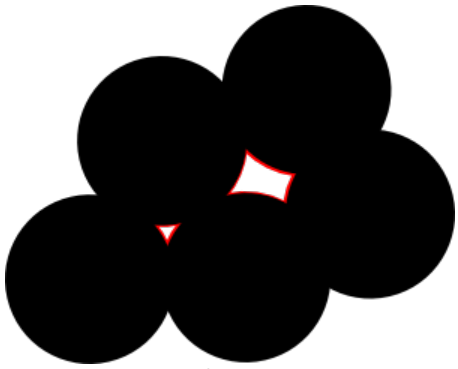
Sample test(s)

input
3 1 2 1 3 1 -1 1
output
3

E. The Last Hole!

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

Luyi has n circles on the plane. The i-th circle is centered at (x_i, y_i). At the time zero circles start to grow simultaneously. In other words, the radius of each circle at time t $(t > 0)$ is equal to t. The circles are drawn as black discs on an infinite white plane. So at each moment the plane consists of several black and white regions. Note that the circles may overlap while growing.



We define a *hole* as a closed, connected white region. For instance, the figure contains two holes shown by red border. During growing some holes may be created and it is easy to see that each created hole will disappear eventually. Luyi asks you to find moment of time such that the last hole disappears. In other words, you should find the first moment such that no hole can be seen after that.

Input

The first line of the input contains integer n $(1 \le n \le 100)$. Each of the next n lines contains two integers x_i and y_i (- $10^4 \le x_i, y_i \le 10^4$), indicating the location of i-th circle.

It's guaranteed that no two circles are centered at the same point.

Output

Print the moment where the last hole disappears. If there exists no moment in which we can find holes print `-1`.

The answer will be considered correct if the absolute or relative error does not exceed 10^{-4}.

Sample test(s)

input
3 0 0 1 1 2 2
output
-1

input
4 0 0 0 2 2 2 2 0
output
1.414214

input
4 0 1 0 -1 -2 0 4 0
output
2.125000

Codeforces (c) Copyright 2010-2015 Mike Mirzayanov
The only programming contests Web 2.0 platform