Codeforces Round #242 (Div. 2)

A. Squats

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Pasha has many hamsters and he makes them work out. Today, n hamsters (n is even) came to work out. The hamsters lined up and each hamster either sat down or stood up.

For another exercise, Pasha needs exactly $\frac{n}{2}$ hamsters to stand up and the other hamsters to sit down. In one minute, Pasha can make some hamster ether sit down or stand up. How many minutes will he need to get what he wants if he acts optimally well?

Input

The first line contains integer n ($2 \le n \le 200$; n is even). The next line contains n characters without spaces. These characters describe the hamsters' position: the i-th character equals 'X', if the i-th hamster in the row is standing, and 'x', if he is sitting.

Output

In the first line, print a single integer — the minimum required number of minutes. In the second line, print a string that describes the hamsters' position after Pasha makes the required changes. If there are multiple optimal positions, print any of them.

Examples

input
4 xxXx
output
1 XxXx

input
2 XX
output
1 xX

input
6 xXXxXx
output
0 xXXxXx

B. Megacity

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

The administration of the Tomsk Region firmly believes that it's time to become a megacity (that is, get population of one million). Instead of improving the demographic situation, they decided to achieve its goal by expanding the boundaries of the city.

The city of Tomsk can be represented as point on the plane with coordinates $(0; 0)$. The city is surrounded with n other locations, the i-th one has coordinates (x_i, y_i) with the population of k_i people. You can widen the city boundaries to a circle of radius r. In such case all locations inside the circle and on its border are included into the city.

Your goal is to write a program that will determine the minimum radius r, to which is necessary to expand the boundaries of Tomsk, so that it becomes a megacity.

Input

The first line of the input contains two integers n and s ($1 \le n \le 10^3$; $1 \le s < 10^6$) — the number of locatons around Tomsk city and the population of the city. Then n lines follow. The i-th line contains three integers — the x_i and y_i coordinate values of the i-th location and the number k_i of people in it ($1 \le k_i < 10^6$). Each coordinate is an integer and doesn't exceed 10^4 in its absolute value.

It is guaranteed that no two locations are at the same point and no location is at point $(0; 0)$.

Output

In the output, print "-1" (without the quotes), if Tomsk won't be able to become a megacity. Otherwise, in the first line print a single real number — the minimum radius of the circle that the city needs to expand to in order to become a megacity.

The answer is considered correct if the absolute or relative error don't exceed 10^{-6}.

Examples

input
4 999998 1 1 1 2 2 1 3 3 1 2 -2 1
output
2.8284271

input
4 999998 1 1 2 2 2 1 3 3 1 2 -2 1
output
1.4142136

input
2 1 1 1 999997 2 2 1
output
-1

C. Magic Formulas

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

People in the Tomskaya region like magic formulas very much. You can see some of them below.

Imagine you are given a sequence of positive integer numbers $p_1, p_2, ..., p_n$. Lets write down some magic formulas:

$$q_i = p_i \oplus (i \bmod 1) \oplus (i \bmod 2) \oplus \cdots \oplus (i \bmod n);$$

$$Q = q_1 \oplus q_2 \oplus \ldots \oplus q_n.$$

Here, "mod" means the operation of taking the residue after dividing.

The expression $x \oplus y$ means applying the bitwise *xor* (excluding "OR") operation to integers x and y. The given operation exists in all modern programming languages. For example, in languages C++ and Java it is represented by "^", in Pascal — by "xor".

People in the Tomskaya region like magic formulas very much, but they don't like to calculate them! Therefore you are given the sequence p, calculate the value of Q.

Input

The first line of the input contains the only integer n ($1 \le n \le 10^6$). The next line contains n integers: $p_1, p_2, ..., p_n$ ($0 \le p_i \le 2 \cdot 10^9$).

Output

The only line of output should contain a single integer — the value of Q.

Examples

input
3 1 2 3
output
3

D. Biathlon Track

time limit per test: 4.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently an official statement of the world Olympic Committee said that the Olympic Winter Games 2030 will be held in Tomsk. The city officials decided to prepare for the Olympics thoroughly and to build all the necessary Olympic facilities as early as possible. First, a biathlon track will be built.

To construct a biathlon track a plot of land was allocated, which is a rectangle divided into $n \times m$ identical squares. Each of the squares has two coordinates: the number of the row (from 1 to n), where it is located, the number of the column (from 1 to m), where it is located. Also each of the squares is characterized by its height. During the sports the biathletes will have to move from one square to another. If a biathlete moves from a higher square to a lower one, he makes a descent. If a biathlete moves from a lower square to a higher one, he makes an ascent. If a biathlete moves between two squares with the same height, then he moves on flat ground.

The biathlon track should be a border of some rectangular area of the allocated land on which biathletes will move in the clockwise direction. It is known that on one move on flat ground an average biathlete spends t_p seconds, an ascent takes t_u seconds, a descent takes t_d seconds. The Tomsk Administration wants to choose the route so that the average biathlete passes it in as close to t seconds as possible. In other words, the difference between time t_s of passing the selected track and t should be minimum.

For a better understanding you can look at the first sample of the input data. In this sample $n = 6, m = 7$, and the administration wants the track covering time to be as close to $t = 48$ seconds as possible, also, $t_p = 3$, $t_u = 6$ and $t_d = 2$. If we consider the rectangle shown on the image by arrows, the average biathlete can move along the boundary in a clockwise direction in exactly 48 seconds. The upper left corner of this track is located in the square with the row number 4, column number 3 and the lower right corner is at square with row number 6, column number 7.

Among other things the administration wants all sides of the rectangle which boundaries will be the biathlon track to consist of no less than three squares and to be completely contained within the selected land.

You are given the description of the given plot of land and all necessary time values. You are to write the program to find the most suitable rectangle for a biathlon track. If there are several such rectangles, you are allowed to print any of them.

Input

The first line of the input contains three integers n, m and t ($3 \le n, m \le 300$, $1 \le t \le 10^9$) — the sizes of the land plot and the desired distance covering time.

The second line also contains three integers t_p, t_u and t_d ($1 \le t_p, t_u, t_d \le 100$) — the time the average biathlete needs to cover a flat piece of the track, an ascent and a descent respectively.

Then n lines follow, each line contains m integers that set the heights of each square of the given plot of land. Each of the height values is a positive integer, not exceeding 10^6.

Output

In a single line of the output print four positive integers — the number of the row and the number of the column of the upper left corner and the number of the row and the number of the column of the lower right corner of the rectangle that is chosen for the track.

Examples

input
6 7 48 3 6 2 5 4 8 3 3 7 9 4 1 6 8 7 1 1 1 6 4 6 4 8 6 7 2 6 1 6 9 4 1 9 8 6 3 9 2 4 5 6 8 4 3 7
output
4 3 6 7

E. Colored Jenga

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Cold winter evenings in Tomsk are very boring — nobody wants be on the streets at such a time. Residents of Tomsk while away the time sitting in warm apartments, inventing a lot of different games. One of such games is 'Colored Jenga'.

This game requires wooden blocks of three colors: red, green and blue. A tower of n levels is made from them. Each level consists of three wooden blocks. The blocks in each level can be of arbitrary colors, but they are always located close and parallel to each other. An example of such a tower is shown in the figure.

The game is played by exactly one person. Every minute a player throws a special dice which has six sides. Two sides of the dice are green, two are blue, one is red and one is black. The dice shows each side equiprobably.

If the dice shows red, green or blue, the player must take any block of this color out of the tower at this minute so that the tower doesn't fall. If this is not possible, the player waits until the end of the minute, without touching the tower. He also has to wait until the end of the minute without touching the tower if the dice shows the black side. **It is not allowed to take blocks from the top level of the tower (whether it is completed or not)**.

Once a player got a block out, he must put it on the top of the tower so as to form a new level or finish the upper level consisting of previously placed blocks. The newly constructed levels should have all the same properties as the initial levels. **If the upper level is not completed, starting the new level is prohibited**.

For the tower not to fall, in each of the levels except for the top, there should be at least one block. Moreover, if at some of these levels there is exactly one block left and this block is not the middle block, the tower falls.

The game ends at the moment when there is no block in the tower that you can take out so that the tower doesn't fall.

Here is a wonderful game invented by the residents of the city of Tomsk. I wonder for how many minutes can the game last if the player acts optimally well? If a player acts optimally well, then at any moment he tries to choose the block he takes out so as to minimize the expected number of the game duration.

Your task is to write a program that determines the expected number of the desired amount of minutes.

Input

The first line of the input contains the only integer n ($2 \le n \le 6$) — the number of levels in the tower.

Then n lines follow, describing the levels of the tower from the bottom to the top (the first line is the top of the tower). Each level is described by three characters, the first and the third of them set the border blocks of the level and the second one is the middle block. The character that describes the block has one of the following values 'R' (a red block), 'G' (a green block) and 'B' (a blue block).

Output

In the only line of the output print the sought mathematical expectation value. The answer will be considered correct if its relative or absolute error doesn't exceed 10^{-6}.

Examples

input
6 RGB GRG BBB GGR BRG BRB
output
17.119213696601992