,

Problem A. Candies

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

Monocarp works as a principal in a school. He has n candies and he wants to gift some of them to his students. There are a boys and b girls in the school.

Monocarp has decided that he'll gift some **positive** number of candies to each boy (the same number of candies for each boy) and some **positive** number of candies to each girl (the same number of candies for each girl). Moreover, the number of candies he gifts to each boy should be **strictly less** than the number of candies he gifts to each girl.

Monocarp wants to gift as many of his n candies as possible. Your task is to calculate the minimum possible number of candies that are not gifted to anyone, if you can decide how many candies each boy will receive and how many candies each girl will receive.

You may assume that the input data meets the following constraints: it is possible to give at least one candy to each boy and at least two candies to each girl (in other words, $a + 2 \cdot b \le n$).

Input

The first line contains three integers n, a and b ($3 \le n \le 1\,000$, $1 \le a, b \le 100$, $a + b \cdot 2 \le n$) — the number of candies Monocarp has, the number of boys and the number of girls, respectively.

Output

Print the minimum possible number of candies that are not gifted to anyone, if you can decide how many candies will receive each boy and how many candies will receive each girl.

Examples

standard input	standard output
34 5 6	0
42 4 7	2

Note

In the first example, Monocarp can give 2 candies to every boy and 4 candies to every girl, so he'll gift all his candies.

In the second example, Monocarp can give 3 candies to every boy and 4 candies to every girl. Then he'll gift $3 \cdot 4 + 4 \cdot 7 = 40$ candies in total, and 2 candies will remain.

Problem B. Computer Game

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Monocarp is playing a computer game. Now he wants to complete the first level of this game.

A level is a rectangular grid of 2 rows and n columns. Monocarp controls a character, which starts in cell $(1, 1)$ — at the intersection of the 1-st row and the 1-st column.

Monocarp's character can move from one cell to another in one step if the cells are adjacent by side and/or corner. Formally, it is possible to move from cell (x_1, y_1) to cell (x_2, y_2) in one step if $|x_1 - x_2| \le 1$ and $|y_1 - y_2| \le 1$. Obviously, it is prohibited to go outside the grid.

There are traps in some cells. If Monocarp's character finds himself in such a cell, he dies, and the game ends.

To complete a level, Monocarp's character should reach cell $(2, n)$ — at the intersection of row 2 and column n.

Help Monocarp determine if it is possible to complete the level.

Input

The first line contains a single integer n ($3 \le n \le 100$) — the number of columns.

Next two lines describe the level. The i-th of these lines describes the i-th line of the level — the line consists of the characters '0' and '1'. The character '0' corresponds to a safe cell, the character '1' corresponds to a trap cell.

An additional constraint on the input: cells $(1, 1)$ and $(2, n)$ are safe.

Output

Output YES if it is possible to complete the level, and NO otherwise.

Examples

standard input	standard output
3 000 000	YES
4 0011 1100	YES
4 0111 1110	NO
6 010101 101010	YES

Note

In the first example, one of the possible paths is $(1, 1) \to (2, 2) \to (2, 3)$.

In the second example, one of the possible paths is $(1, 1) \to (1, 2) \to (2, 3) \to (2, 4)$.

In the fourth example, one of the possible paths is $(1, 1) \to (2, 2) \to (1, 3) \to (2, 4) \to (1, 5) \to (2, 6)$.

Problem C. Groups

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

n students attended the first meeting of the Berland SU programming course (n is even). All students will be divided into two groups. Each group will be attending exactly one lesson each week during one of the five working days (Monday, Tuesday, Wednesday, Thursday and Friday), and the days chosen for the groups must be different. Furthermore, both groups should contain the same number of students.

Each student has filled a survey in which they told which days of the week are convenient for them to attend a lesson, and which are not.

Your task is to determine if it is possible to choose two different week days to schedule the lessons for the group (the first group will attend the lesson on the first chosen day, the second group will attend the lesson on the second chosen day), and divide the students into two groups, so the groups have equal sizes, and for each student, the chosen lesson day for their group is convenient.

Input

The first line contains one integer n ($2 \le n \le 1\,000$) — the number of students.

The i-th of the next n lines contains 5 integers, each of them is 0 or 1. If the j-th integer is 1, then the i-th student can attend the lessons on the j-th day of the week. If the j-th integer is 0, then the i-th student cannot attend the lessons on the j-th day of the week.

Additional constraint on the input: for each student, at least one of the days of the week is convenient.

Output

If it's possible to divide the students into two groups of equal sizes and choose different days for the groups so each student can attend the lesson in the chosen day of their group, print "YES" (without quotes). Otherwise, print "NO" (without quotes).

Examples

standard input	standard output
4 1 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0	YES
2 0 0 0 1 0 0 0 0 1 0	NO

Note

In the first example, there is a way to meet all the constraints. For example, the first group can consist of the first and the third students, they will attend the lessons on Thursday (the fourth day); the second group can consist of the second and the fourth students, and they will attend the lessons on Tuesday (the second day).

In the second example, it is impossible to divide the students into groups so they attend the lessons on different days.

Problem D. Rectangle Restoration

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

In this problem, you have to restore some data about a rectangle. You know that the sum of lengths of some **two** sides of this rectangle is equal to x, and the sum of lengths of some **three** sides of this rectangle is equal to y. Note that each of the four sides of rectangle is included at most once into each of the two sums.

You have to calculate the minimum possible perimeter of a rectangle such that the sum of lengths of some two sides of this rectangle is equal to x, and the sum of lengths of some three sides of this rectangle is equal to y. Note that side lengths are not necessarily integers, but they are **strictly positive**.

Input

The only line contains two integers x and y ($1 \le x, y \le 10^9$).

Output

If there is no rectangle meeting the constraints, print -1.

Otherwise, print one real number — the minimum possible perimeter of a rectangle meeting the constraints. The absolute or relative error of your answer must not exceed 10^{-4}.

Examples

standard input	standard output
10 15	20.0000
6 4	7.0000
10 2	-1
7 4	7.5000
500000000 1000000000	1250000000.0

Note

In the first example, the rectangle in the answer is a square with side length equal 5.

In the second example, the rectangle with minimum perimeter meeting the constraints has two sides with length 3 each, and two sides with length 0.5 each.

In the third example, there is no rectangle meeting the constraints.

In the fourth example, the rectangle with minimum perimeter meeting the constraints has two sides with length 3.5, each, and two sides with length 0.25 each.

Problem E. Delete Two Elements

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Monocarp has got an array a consisting of n integers. Let's denote k as the mathematic mean of these elements (note that it's possible that k is not an integer).

The mathematic mean of an array of n elements is the sum of elements divided by the number of these elements (i. e. sum divided by n).

Monocarp wants to delete exactly two elements from a so that the mathematic mean of the remaining $(n-2)$ elements is still equal to k.

Your task is to calculate the number of pairs of positions $[i, j]$ $(i < j)$ such that if the elements on these positions are deleted, the mathematic mean of $(n-2)$ remaining elements is equal to k (that is, it is equal to the mathematic mean of n elements of the original array a).

Input

The first line contains one integer n $(3 \le n \le 200\,000)$ — the number of elements in the array.

The second line contains a sequence of integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$, where a_i is the i-th element of the array.

Output

Print one integer — the number of pairs of positions $[i, j]$ $(i < j)$ such that if the elements on these positions are deleted, the mathematic mean of $(n-2)$ remaining elements is equal to k (that is, it is equal to the mathematic mean of n elements of the original array a).

Examples

standard input	standard output
4 8 8 8 8	6
3 50 20 10	0
5 1 4 7 3 5	2

Note

In the first example, any pair of elements can be removed since all of them are equal.

In the second example, there is no way to delete two elements so the mathematic mean doesn't change.

In the third example, it is possible to delete the elements on positions 1 and 3, or the elements on positions 4 and 5.

Problem F. Coconuts

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Monocarp is stranded on a deserted island! After exploring the island, he found out that n palms grow on it, and there are a_i coconuts on the i-th palm.

Monocarp has decided to choose a positive integer x and gather coconuts from each palm such that the number of coconuts on that palm is an integer positive power x. In other words, Monocarp will gather all of the coconuts from the i-th palm, if a_i can be represented as x^m, where m is a positive integer.

Your task is to help Monocarp choose x optimally and gather the maximum possible number of coconuts.

Input

The first line contains one integer n ($2 \le n \le 200\,000$) — the number of palms.

The second line contains a sequence of integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), where a_i is the number of coconuts growing on the i-th palm.

Output

Print one integer — the maximum number of coconuts Monocarp can gather if he chooses x optimally.

Examples

standard input	standard output
5 4 8 25 5 16	30
3 9 27 40	40
6 1 1 4 1 1 1	5

Note

In the first example, Monocarp should choose $x = 5$. Then he can gather the coconuts from the third palm and the fourth palm, getting $25 + 5 = 30$ coconuts in total.

In the second example, Monocarp should choose $x = 40$. Then he can gather 40 coconuts from the third palm. If Monocarp chooses, for example, 3, he can gather the coconuts from the first two palms, but then he will get only $9 + 27 = 36$ coconuts which is less than the result for $x = 40$.

In the third example, Monocarp should choose $x = 1$. Then he can gather 5 coconuts.

Problem G. Training Session

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Monocarp is the coach of the Berland State University programming teams. He decided to compose a problemset for a training session for his teams.

Monocarp has n problems that none of his students have seen yet. The i-th problem has a topic a_i (an integer from 1 to 200 000) and a difficulty b_i (an integer from 1 to 200 000). All problems are different, that is, there are no two problems that have the same topic and difficulty at the same time.

Monocarp decided to select exactly 3 problems from n problems for the problemset. The problems should satisfy **at least one** of two conditions (possibly, both):

- the topics of all three selected problems are different;
- the difficulties of all three selected problems are different.

Your task is to determine the number of ways to select three problems for the problemset.

Input

The first line contains an integer n ($3 \le n \le 200\,000$) — the number of problems that Monocarp has.

In the i-th of the following n lines, there are two integers a_i and b_i ($1 \le a_i, b_i \le 200\,000$) — the topic and the difficulty of the i-th problem.

It is guaranteed that there are no two problems that have the same topic and difficulty at the same time.

Output

Print the number of ways to select three training problems that meet either of the requirements described in the statement.

Examples

standard input	standard output
4 2 4 3 4 2 1 1 5	3
5 1 5 2 4 3 3 4 2 5 1	10

Note

In the first example, you can take the following sets of three problems:

- problems 1, 2, 4;
- problems 1, 3, 4;

,

- problems 2, 3, 4.

Thus, the number of ways is equal to three.

,

Problem H. Colored Balls

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

Monocarp has got several colored balls — a red ones, b green ones and c blue ones.

Monocarp can transform the balls as follows: he can take two balls of different colors and merge them into a ball of the third color (two balls he has taken disappear, and a new ball of the third color appears). For example, Monocarp can merge a red ball and a blue ball into a green ball.

Monocarp wants to perform several (maybe zero) operations in such a way that, after performing them, the number of balls of each color is the same for all colors. Your task is to calculate if it is possible, and determine the minimum number of operations Monocarp has to perform to reach his goal (if he can do it at all).

Input

The first line contains three integers a, b and c ($0 \le a, b, c \le 10^9; a + b + c > 0$) — the number of red, green and blue balls Monocarp has.

Output

If Monocarp cannot reach his goal, print -1.

Otherwise, print one integer — the minimum number of operations Monocarp has to perform.

Examples

standard input	standard output
3 3 1	1
101 101 101	0
15 30 20	-1
12 74 58	39

Note

In the first example, Monocarp can merge a red ball and a green ball to get a blue ball. Then he will have two balls for each color.

In the second example, Monocarp doesn't have to perform any operations.

In the third example, it's impossible to achieve Monocarp's goal.

Problem I. Tree Painting

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Monocarp has got a tree consisting of n vertices, n is even. A tree is a connected acyclic graph.

Monocarp wants to paint exactly half of the vertices of the tree black, and the other half of the vertices — white. Furthermore, he wants there to be exactly one edge connecting vertices of different colors after all vertices are painted.

Help Monocarp find any coloring that meets these constraints, or report that it is impossible.

Input

The first line contains one even integer n ($2 \le n \le 200\,000$) — the number of vertices.

Each of the next $n - 1$ lines contains two integers u and v ($1 \le u, v \le n$) denoting the endpoints of some edge. It is guaranteed that these edges form a tree.

Output

If there is no coloring that meets the constraints, print "NO" (without quotes).

Otherwise, print "YES" in the first line (without quotes). Then, print n integers 0 and/or 1 in the second line, where the i-th integer should be 0 if the i-th vertex should be painted black, or the i-th integer should be 1 if the i-th vertex should be painted white. If there are multiple answers, print any of them.

Examples

standard input	standard output
6 1 2 2 3 4 2 4 6 5 4	YES 0 0 0 1 1 1
4 3 1 3 2 3 4	NO

Note

In the first example, you should paint the vertices 1, 2 and 3 white, and vertices 4, 5 and 6 — black. Then the only edge connecting two vertices of different colors is the edge between 2 and 4.

There is no coloring that meets the constraints in the second example.

Problem J. Replacing Letters

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Monocarp has got a string s consisting of lowercase Latin letters.

Monocarp likes *non-descending* strings. A string is called non-descending if for each pair of positions $[i, j]$ where $i < j$ the following condition is met: the i-th letter of the string appears not later in the alphabet than the j-th letter of the string. For example, the strings "aaa", "bcdef", "aappszz" are non-descending, but the strings "ba", "abcba", "zzzzzx" are not

Monocarp wants his string s to become non-descending. To achieve his goal, he can replace any letters in the string with any other letters. Calculate the minimum number of letters in s that should be replaced to make s non-descending, and print the resulting string s.

Input

The first line contains one integer n ($2 \le n \le 200\,000$) — the length of s.

The second line contains the string s consisting of n lowercase Latin letters.

Output

In the first line, print one integer x — the minimum number of letters that should be replaced in s so it becomes non-descending.

In the second line, print the resulting string — a non-descending string that can be obtained from s by replacing x letters. If there are multiple answers, print any of them.

Examples

standard input	standard output
5 efahi	1 efghi
10 aaaabceehh	0 aaaabceehh
8 fgdadvfd	5 ddddddddd

Problem K. Staircases

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a matrix, consisting of n rows and m columns.

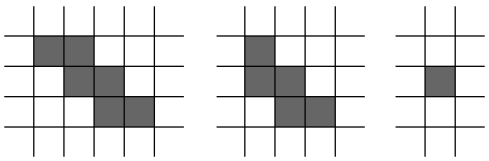
Each cell of the matrix can be either free or locked.

Let's call a path in the matrix *a staircase* if it:

- starts and ends in the free cell;
- visits only free cells;
- has one of the two following structures:
  1. the second cell is 1 to the right from the first one, the third cell is 1 to the bottom from the second one, the fourth cell is 1 to the right from the third one, and so on;
  2. the second cell is 1 to the bottom from the first one, the third cell is 1 to the right from the second one, the fourth cell is 1 to the bottom from the third one, and so on.

In particular, a path, consisting of a single cell, is considered to be a staircase.

Here are some examples of staircases:



Initially all the cells of the matrix are **free**.

You have to process q queries, each of them flips the state of a single cell. So, if a cell is currently free, it makes it locked, and if a cell is currently locked, it makes it free.

Print the number of different staircases after each query. Two staircases are considered different if there exists such a cell that appears in one path and doesn't appear in the other path.

Input

The first line contains three integers n, m and q ($1 \le n, m \le 1000$; $1 \le q \le 10^4$) — the sizes of the matrix and the number of queries.

Each of the next q lines contains two integers x and y ($1 \le x \le n$; $1 \le y \le m$) — the description of each query.

Output

Print q integers — the i-th value should be equal to the number of different staircases after i queries. Two staircases are considered different if there exists such a cell that appears in one path and doesn't appear in the other path.

Examples

standard input	standard output
2 2 8 1 1 1 1 1 1 2 2 1 1 1 2 2 1 1 1	5 10 5 2 5 3 1 0
3 4 10 1 4 1 2 2 3 1 2 2 3 3 2 1 3 3 4 1 3 3 1	49 35 24 29 49 39 31 23 29 27
1000 1000 2 239 634 239 634	1332632508 1333333000

Problem L. RBS

Input file:	standard input
Output file:	standard output
Time limit:	3 seconds
Memory limit:	512 megabytes

A bracket sequence is a string containing only characters "(" and ")". A regular bracket sequence (or, shortly, an RBS) is a bracket sequence that can be transformed into a correct arithmetic expression by inserting characters "1" and "+" between the original characters of the sequence. For example:

- bracket sequences "()()" and "(())" are regular (the resulting expressions are: "(1)+(1)" and "((1+1)+1)");
- bracket sequences ")(", "(" and ")" are not.

Let's denote the concatenation of two strings x and y as $x + y$. For example, "()()" + ")(" = "()())(".

You are given n bracket sequences $s_1, s_2, \ldots, s_n$. You can rearrange them in any order (you can rearrange only the strings themselves, but not the characters in them).

Your task is to rearrange the strings in such a way that the string $s_1 + s_2 + \cdots + s_n$ has as many non-empty prefixes that are RBS as possible.

Input

The first line contains a single integer n ($1 \le n \le 20$).

Then n lines follow, the i-th of them contains s_i — a bracket sequence (a string consisting of characters "(" and/or ")". All sequences s_i are non-empty, their total length does not exceed $4 \cdot 10^5$.

Output

Print one integer — the maximum number of non-empty prefixes that are RBS for the string $s_1 + s_2 + \cdots + s_n$, if the strings $s_1, s_2, \ldots, s_n$ can be rearranged arbitrarily.

Examples

standard input	standard output
2 ( )	1
4 ()()()) (( )	4
1 (())	1
1)(()	0

Note

In the first example, you can concatenate the strings as follows: "(" + ")" = "()", the resulting string will have one prefix, that is an RBS: "()".

,

In the second example, you can concatenate the strings as follows: “(” + “)” + “()()())” + “(” = “()()()())(”, the resulting string will have four prefixes that are RBS: “()”, “()()”, “()()()”, “()()()()”.

The third and the fourth examples contain only one string each, so the order is fixed.

Problem M. The Sum of Good Numbers

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

Let's call a positive integer *good* if there is no digit 0 in its decimal representation.

For an array of a *good* numbers a, one found out that the sum of some two neighboring elements is equal to x (i.e. $x = a_i + a_{i+1}$ for some i). x had turned out to be a *good* number as well.

Then the elements of the array a were written out one after another without separators into one string s. For example, if $a = [12, 5, 6, 133]$, then $s = 1256133$.

You are given a string s and a number x. Your task is to determine the positions in the string that correspond to the adjacent elements of the array that have sum x. If there are several possible answers, you can print any of them.

Input

The first line contains the string s ($2 \le |s| \le 5 \cdot 10^5$).

The second line contains an integer x ($2 \le x < 10^{200000}$).

An additional constraint on the input: the answer always exists, i.e you can always select two adjacent substrings of the string s so that if you convert these substrings to integers, their sum is equal to x.

Output

In the first line, print two integers l_1, r_1, meaning that the first term of the sum (a_i) is in the string s from position l_1 to position r_1.

In the second line, print two integers l_2, r_2, meaning that the second term of the sum (a_{i+1}) is in the string s from position l_2 to position r_2.

Examples

standard input	standard output
1256133 17	1 2 3 3
9544715561 525	2 3 4 6
239923 5	1 1 2 2
1218633757639 976272	2 7 8 13

Note

In the first example $s[1;2] = 12$ and $s[3;3] = 5$, $12 + 5 = 17$.

In the second example $s[2;3] = 54$ and $s[4;6] = 471$, $54 + 471 = 525$.

In the third example $s[1;1] = 2$ and $s[2;2] = 3$, $2 + 3 = 5$.

In the fourth example $s[2;7] = 218633$ and $s[8;13] = 757639$, $218633 + 757639 = 976272$.