Problem A. Rook

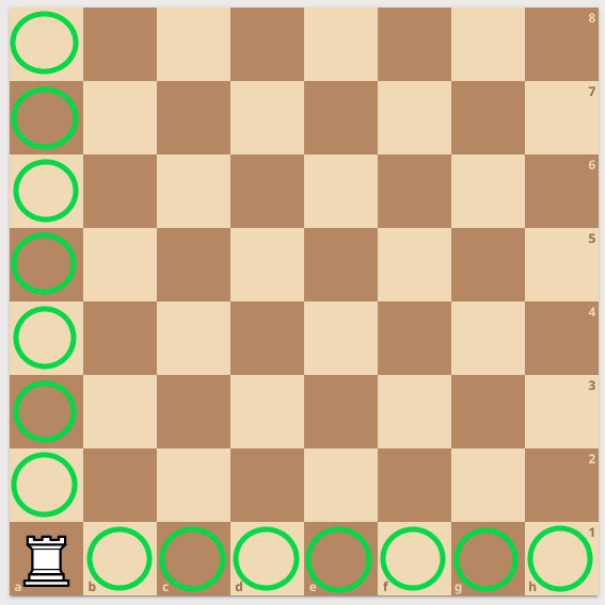
Time limit: 1 second
Memory limit: 256 megabytes

Given a chessboard of size $n \times m$. That is, with n rows and m columns.

On this chessboard, there is only one piece — the rook. It is located in the bottom left corner. There are no other pieces.

Recall that the rook can move any positive number of squares horizontally or vertically in one move, but not diagonally.

Find the number of squares to which the rook can move in exactly one move.



The picture shows a traditional chessboard of size 8×8. On it, the rook can move to all the squares marked in green. There are a total of 14 such squares, so the answer is 14.

Input

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

The second line contains one integer m ($1 \le m \le 20$).

Output

Output the number of squares to which the rook can move in one move.

Examples

standard input	standard output
8 8	14
3 2	3

Note

An explanation of why the answer is 14 for the first example can be seen in the picture above.

In the second example, the answer is 3, because the rook can only move to one position to the right and two positions up.

Problem B. Coordinates

Time limit: 1 second
Memory limit: 256 megabytes

Given a point (x, y, z) in 3D space.

Find the **squared** distance from this point to the origin (i.e., to the point $(0, 0, 0)$).

Recall that the distance between two points (x_1, y_1, z_1) and (x_2, y_2, z_2) is defined by the formula

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Input

The first line contains one integer x ($-100 \le x \le 100$).

The second line contains one integer y ($-100 \le y \le 100$).

The third line contains one integer z ($-100 \le z \le 100$).

Output

Output one integer.

Example

standard input	standard output
1 -3 5	35

Note

In the first test, we are interested in the squared distance from the point $(1, -3, 5)$ to $(0, 0, 0)$.

Substituting the coordinates into the formula, we get:

$$\left(\sqrt{(1-0)^2 + (-3-0)^2 + (5-0)^2}\right)^2 =$$

$$= \left(\sqrt{1 + 9 + 25}\right)^2 = 35$$

Problem C. Cook the Homework

Time limit: 1 second
Memory limit: 256 megabytes

Sakurako is preparing for her next programming lesson, which starts today at 14 : 00, but she needs to finish her unfinished homework. The lesson starts in a minutes, and completing the homework will take b minutes. Fortunately, Sakurako worked on the homework for c minutes yesterday.

Your task is to determine whether Sakurako can finish the remaining homework before the lesson starts.

For example, suppose the lesson starts in 5 minutes, Sakurako's homework requires 6 minutes to complete, and she has already worked on it for 2 minutes. In this case, she needs $6 - 2 = 4$ minutes to finish it. Therefore, if she starts preparing for the lesson right away, she will complete her homework (and even have one extra minute left).

Input

A single line contains three integers a, b, and c ($0 \le a, b, c \le 360$).

Output

Print "`YES`", if Sakurako has a chance to finish her homework before the lesson starts, or "`NO`" otherwise.

Scoring

You will receive a certain number of points if your solution works correctly for $c = 0$; that is, Sakurako did not work on her homework yesterday.

Examples

standard input	standard output
5 6 2	YES
5 8 3	YES
5 8 1	NO
0 0 0	YES

Note

The first example was explained in the legend.

In the second example, the lesson starts in 5 minutes. Sakurako needs $8 - 3 = 5$ minutes to finish her homework. Therefore, she fits perfectly into her schedule.

In the fourth example, the lesson has already started, but the homework requires zero minutes to complete (was it oral homework?).

Problem D. Balls and Bins

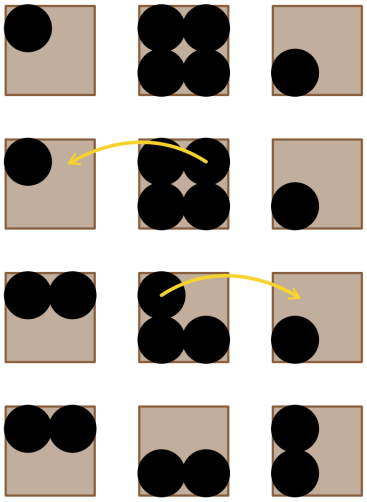
Time limit: 1 second
Memory limit: 256 megabytes

Sakurako is a young programmer working on a robotics project.

She has built three smart bins, each starting with a, b, and c balls, respectively. The bins are designed to pass balls between each other, and her task is to write a program that makes the number of balls in all three bins equal.

Sakurako can instruct a non-empty bin to take one ball out and give it to another. However, she's curious to know if it's even possible to balance the bins, and if so, how many moves it would take. Can you help her design an efficient solution?

Let's say she has three bins; the first bin has 1 ball, the second bin has 4 balls, and the third bin has 1 ball. Below, you can find the illustration for such an example.



Yellow arrows show the ball we take and where we put it. In this case, we perform two operations.

Input

The only line contains three integers a, b, and c ($0 \le a, b, c \le 10^8$).

Output

If it is impossible to make the number of balls in all three bins equal, output "`-1`".

Otherwise, output the minimum number of moves required to balance the balls over the bins.

Scoring

You will receive at least 25 points if your solution works correctly for $b = 0$ and $c = 0$; that is, the second and third bins are empty.

You will receive at least 25 points if your solution works correctly for $b = 2 \cdot a$ and $c = 3 \cdot a$.

Examples

standard input	standard output
1 4 1	2
2 2 3	-1
0 0 0	0
12 0 0	8

Note

In the second example, it might be shown that it is impossible to distribute the balls among the bins as described in the statement.

In the third example, all the bins are empty, which implies that all of them have the same amount of balls; hence, the output is zero.

In the fourth example, we may follow the strategy described below:

$$\begin{array}{lcl} (12,0,0) & \to & (11,1,0) \\ (11,1,0) & \to & (10,1,1) \\ (10,1,1) & \to & (9,2,1) \\ (9,2,1) & \to & (8,2,2) \\ (8,2,2) & \to & (7,3,2) \\ (7,3,2) & \to & (6,4,2) \\ (6,4,2) & \to & (5,4,3) \\ (5,4,3) & \to & (4,4,4) \end{array}$$

what takes us exactly 8 operations. We may prove that it is impossible to balance the bins in less than 8 operations.

Problem E. Sakurako is Late

Time limit: 1 second
Memory limit: 256 megabytes

Another day at school: another time Sakurako is late!

Today she overslept and needs to get to school as fast as possible.

There are n pedestrian crossings that separate Sakurako from her school; each of them contains a single traffic light. Each traffic light is either green or red. The color of each traffic light changes every minute.

Sakurako is not really fast, so she spends one minute in order to cross one pedestrian crossing. Also, she is very law-obedient, so she will not cross the crossing if the light is red. Lastly, she will **always** cross the street if the light is green.

Determine what is the minimum time that Sakurako needs to spend in order to get to the school.

Input

The first line contains one integer n ($1 \le n \le 10^6$), which denotes the number of pedestrian crossings that Sakurako needs to cross.

The second line contains a string s of n characters that represent the initial color of each traffic light; each character is either "`G`" (green) or "`R`" (red). The first character stands for the color of the closest traffic light, and the last character stands for the furthest traffic light.

Output

In a single line output, a single integer which denotes the minimum time needed for Sakurako to cross the crossings.

Scoring

You will receive at least 30 points if your solution works correctly for $s_1 = s_2 = \cdots = s_{n-1} = s_n$; that is, all the colors are the same.

You will receive at least 30 points if your solution works correctly for $s_i \neq s_{i+1}$ ($1 \le i \le n-1$); that is, all adjacent colors are different.

Examples

standard input	standard output
5 RGRRG	7
10 GGRRGRGRRG	13

Note

In the first example, Sakurako may move in a following way:

Time 0:

Sakurako is in position 0.

Time 1:

Sakurako is in position 0 and starts moving across the 1-st pedestrian crossing.

Time 2:

Sakurako is in position 1 and starts moving across the 2-nd pedestrian crossing.

Time 3:

Sakurako is in position 2 and starts moving across the 3-rd pedestrian crossing.

Time 4:

Sakurako is in position 3.

Time 5:

Sakurako is in position 3 and starts moving across the 4-th pedestrian crossing.

Time 6:

Sakurako is in position 4 and starts moving across the 5-th pedestrian crossing.

Time 7:

Sakurako passed all 5 crossings.

In the second example, we can show that the minimal time that Sakurako needs to cross all the crossings is 13.

Problem F. Beautiful String

Time limit: 1 second
Memory limit: 256 megabytes

A binary string is a string which contains only "0" and "1".

A substring of the string is a string that can be obtained by deleting some number of characters from the start and/or end.

Sakurako considers a binary string beautiful if the number of ones is at least 2 and this number divides the length of the string.

For example, "1001", "011100" and "010100100" are beautiful strings, but strings "101", "01010" and "1001000" are not beautiful.

Sakurako loves beautiful strings, but she is bad at searching for them. So, she asks you to help her find any beautiful substring of the given string.

Input

The first line contains one integer n ($3 \le n \le 2 \cdot 10^5$) — the length of the string.

The second line contains one binary string — the string where Sakurako lost her beautiful string.

It is guaranteed that there are at least three characters "1" in the string.

It is possible to show that the answer always exists.

Output

Output two integers l and r ($1 \le l \le r \le n$) where the substring starting at l and ending at r is beautiful. If there are multiple solutions, output any.

Scoring

You will receive at least 32 points if your solution works correctly for $n \le 1000$.

Examples

standard input	standard output
5 10101	1 4
6 101101	3 4

Note

In the first example, the substring "1010" is beautiful since the length, which is 4, is divisible by the number of ones, which is 2.

In the second example, the substring "11" is beautiful since the length, which is 2, is divisible by the number of ones, which is 2.

Problem G. Beautiful Array

Time limit:	1 second
Memory limit:	256 megabytes

Array a is beautiful if all $a_i - a_{i-1}$ $(2 \le i \le n)$ are distinct.

For example, $[1, 3, 2, 9]$ and $[8, 4, 6, 9]$ are beautiful, while $[1, 2, 3, 9]$ and $[1, 5, 5, 9]$ are not.

A sequence of integers $b_1, b_2, \ldots, b_n$ is called a permutation of the sequence $c_1, c_2, \ldots, c_n$ if the number of each unique number in these arrays is the same. For example, $[1, 5, 3, 4, 2]$ is a permutation of $[5, 2, 1, 4, 3]$, and $[1, 3, 3, 5]$ is a permutation of $[3, 5, 1, 3]$. However, $[1, 3, 4]$ is not a permutation of $[2, 3, 4]$ (it doesn't contain 2, which should appear in the sequence and has 1, which should not appear).

Sakurako received an array. She wants to find an example of a permutation of the given array which is beautiful.

Input

The first line contains one integer n $(2 \le n \le 2 \cdot 10^5)$ — the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$ — elements of the array.

It is guaranteed that all numbers in the array are distinct. It is also guaranteed that an answer exists.

Output

Output a beautiful array (which is a permutation of the original array) in one line. If there are multiple solutions, output any.

Scoring

You will receive at least 16 points if your solution works correctly for $n \le 10$.

You will receive at least 48 points if your solution works correctly for $n \le 1000$.

Examples

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

Note

In the first example, array $[3, 5, 4, 1, 2]$ is beautiful since if we take $a_i - a_{i-1}$ for all i, it will be:

- $5 - 3 = 2$;
- $4 - 5 = -1$;
- $1 - 4 = -3$;
- $2 - 1 = 1$.

As we see, all numbers are distinct.

In the second example, array $[4, 2, 3]$ is beautiful since if we take $a_i - a_{i-1}$ for all i, it will be:

- $2 - 4 = -2$;

- $3 - 2 = 1$.

As we see, all numbers are distinct.

Problem H. Sakurako and a Christmas Array

Time limit: 1.5 seconds
Memory limit: 256 megabytes

Sakurako has recently decorated an array in a way that it became a Christmas array. It was **glorious**, until Chefir came and messed it up.

An array of length n is called **glorious** if the following holds:

- for every i $(1 \le i \le n)$ $a_i = a_{n-i+1}$;
- and there is an element which occurs more than $\frac{n}{2}$ times.

You are given an array a of length n, where n is even. You are asked what is the minimum amount of elements that you need to replace in order for this array to become **glorious**.

A "replace" means that you can choose any i $(1 \le i \le n)$ and assign $a_i = x$ where x is an arbitrary integer (even if x does not exist in the array).

Input

The first line contains a single integer n $(2 \le n \le 5 \cdot 10^5$; n is even), which denotes the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^6)$.

Output

In a single line, output a single integer, which denotes the minimum amount of elements that need to be replaced in order for a to become a **glorious** array.

Scoring

You will receive at least 10 points if your solution works correctly for $a_i = a_{i+1}$ for every $(1 \le i < n)$.

You will receive at least 20 points if your solution works correctly for $a_i = a_{n-i+1}$ for every i $(1 \le i \le n)$.

You will receive at least 30 points if your solution works correctly for $n \le 1000$.

You will receive additional 40 points if your solution works correctly without any restrictions.

Examples

standard input	standard output
6 2 6 4 6 2 9	3
6 3 7 2 4 9 3	3

Note

Optimal replacement sequence in test case 1:

$[2, 6, 4, 6, 2, 9] \to [2, 2, 6, 6, 2, 2]$ For this array both conditions hold:

- for each i $(1 \le i \le n)$ $a_i = a_{n-i+1}$
- integer 2 occurs more than $\frac{6}{2} = 3$ times in our array.

We can show that this is the minimal amount of operations that we need to make.

In test case 2 the optimal replacement is:

$[3, 7, 2, 4, 9, 3] \to [3, 4, 4, 4, 4, 3]$

It can be shown that this is the smallest amount of operations that we need to do.