A. Prefix and Suffix Mex

Limits: 2 sec., 512 MiB

For a sequence X composed of a finite number of non-negative integers, we define $\text{mex}(X)$ as the smallest non-negative integer not in X. For example, $\text{mex}(0, 0, 1, 3) = 2, \text{mex}(1) = 0, \text{mex}() = 0$.

You are given two sequences b and c of n non-negative integers.

Construct a sequence a of n non-negative integers that satisfies all of the following conditions.

- $\text{mex}(a_1, a_2, \ldots, a_i) = b_i$ for all $1 \le i \le n$.
- $\text{mex}(a_i, a_{i+1}, \ldots, a_n) = c_i$ for all $1 \le i \le n$.

It is guaranteed that for a given input such a sequence exists.

Input

The first line contains an integer n.
The second line contains n integers b_i.
The third line contains n integers c_i.

Output

Print n integers a_i – the elements of sequence a. The values a_i must not be greater than 10^9. One can prove that there exists a sequence satisfying this constraint.

If there are multiple answers, any of them will be accepted.

Constraints

$1 \le n \le 10^5$,
$0 \le b_i, c_i \le n$,
it is guaranteed that there exists a sequence a satisfying the conditions.

Samples

Input (*stdin*)	Output (*stdout*)
4 1 1 1 3 3 3 2 0	0 2 0 1
7 0 0 3 4 4 4 4 4 4 4 4 3 2 0	1 2 0 3 2 0 1
4 0 0 0 0 0 0 0 0	4 7 74 474747474

Notes

In the first example, we have $b = (1, 1, 1, 3), c = (3, 3, 2, 0)$. Let us show that the sequence $a = (0, 2, 0, 1)$ is a possible answer.

- $\text{mex}(a_1) = \text{mex}(0) = 1 = b_1$.
- $\text{mex}(a_1, a_2) = \text{mex}(0, 2) = 1 = b_2$.
- $\text{mex}(a_1, a_2, a_3) = \text{mex}(0, 2, 0) = 1 = b_3$.
- $\text{mex}(a_1, a_2, a_3, a_4) = \text{mex}(0, 2, 0, 1) = 3 = b_4$.
- $\text{mex}(a_1, a_2, a_3, a_4) = \text{mex}(0, 2, 0, 1) = 3 = c_1$.
- $\text{mex}(a_2, a_3, a_4) = \text{mex}(2, 0, 1) = 3 = c_2$.
- $\text{mex}(a_3, a_4) = \text{mex}(0, 1) = 2 = c_3$.
- $\text{mex}(a_4) = \text{mex}(1) = 0 = c_4$.

Thus, since all the conditions are met, $a = (0, 2, 0, 1)$ is an answer to the problem.

B. Graph on a Plane

Limits: 4 sec., 512 MiB

You are given n unique points (x_i, y_i) on a two-dimensional plane.

A weighted undirected graph G with real-valued weights is *good* if it satisfies the following conditions.

- G contains n vertices numbered 1 through n.
- For each pair of vertices (i, j), the shortest-path weight between vertices i and j in G equals the Euclidean distance between points (x_i, y_i) and (x_j, y_j).

Find the minimum number of edges in a good graph.

Input

The first line contains an integer n – the number of points.

The following n lines contain two integers x_i and y_i – coordinates of the points.

Output

In a single line, print an integer – the minimum number of edges in a good graph.

Constraints

$1 \le n \le 2000$,

$|x_i|, |y_i| \le 10^9$.

Samples

Input (*stdin*)	Output (*stdout*)
4 0 0 4 3 4 2 0 4	6
5 0 0 1 0 0 2 -3 0 0 -4	8

Notes

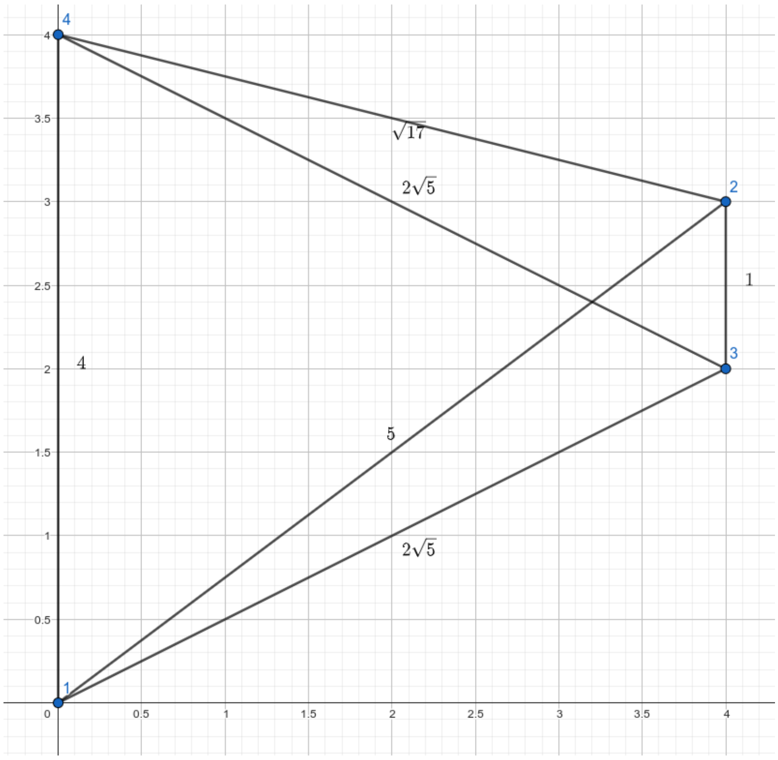


Figure 1: In the first example, the minimum number of edges in a good graph is six.

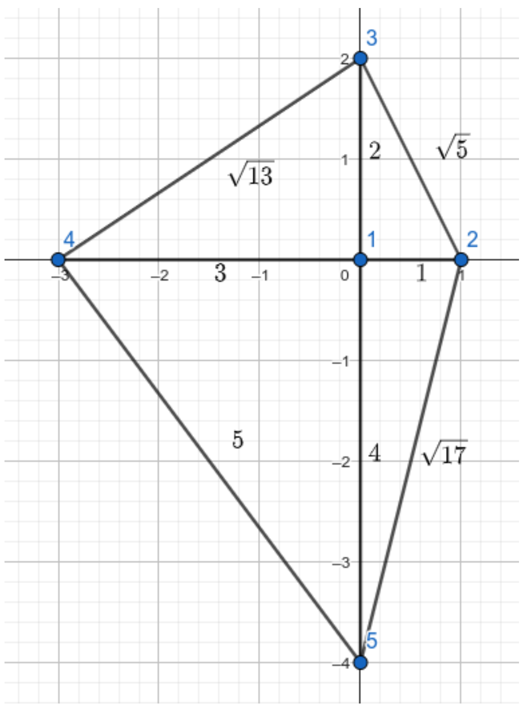


Figure 2: In the second example, the minimum number of edges in a good graph is eight.

C. Divisible Array

Limits: 2 sec., 512 MiB

You are given a sequence a of n non-negative integers.

Determine if there exists a sequence b that is a permutation of a such that b_i is a multiple of $(b_{i+1} + b_{i+2})$ for all $1 \le i \le n - 2$.

Input

The first line contains an integer n – the number of elements in sequence a.

The second line contains n integers a_i – the elements of sequence a.

Output

If there exists a sequence b satisfying the condition, print `Yes`. Otherwise, print `No`.

Constraints

$3 \le n \le 10^6$,
$1 \le a_i \le 10^9$.

Samples

Input (*stdin*)	Output (*stdout*)
`4` `40 4444 4 4`	`Yes`
`7` `4 7 77 4 477 4747 777444777`	`No`

Notes

In the first example, the sequence $b = (4444, 40, 4, 4)$ satisfies the condition.

- $b_1 = 4444$ is a multiple of $b_2 + b_3 = 40 + 4 = 44$.
- $b_2 = 40$ is a multiple of $b_3 + b_4 = 4 + 4 = 8$.

In the second example, there does not exist a sequence satisfying the condition.

D. Minimum Queries

Limits: 2 sec., 512 MiB

This is an ***interactive task****, where your program and the judge interact via standard input and output.*

The judge has a permutation p of integers $(1, 2, \ldots, n)$.

You are given two integers n and m. The permutation p is not given to you.

You can ask the judge at most $n - 1$ following questions.

- Choose two integers l and r such that $1 \le l \le r \le n$ and ask the index i such that $l \le i \le r$ and $p_i \le p_j$ for all $l \le j \le r$. In other words, ask the position of the minimum element in the range $p[l, r]$.

Find the index k such that $a_k = m$. If it is impossible to find it, report this.

Input

First, receive two integers n and m from standard input.

Then, you get to ask the judge at most $n - 1$ questions as described in the problem statement.

Print each question to standard output in the format "`?` l r", where l, r are integers satisfying $1 \le l \le r \le n$.

In response to this, the answer to your question – the index i – will be given from standard input. Here, $l \le i \le r$.

Output

When you find the index k satisfying $a_k = m$, print it in the format "`!` k".

If it is impossible to find this index, print `! -1`.

Constraints

$1 \le m \le n \le 10^4$.

Notes

Print a newline and flush standard output at the end of each message.

After printing the answer, immediately quit the program.

The permutation p will be fixed at the start of the interaction and will not be changed according to your questions or other factors.

In the following interaction, $n = 7$, $m = 3$, and $p = (1, 3, 5, 6, 4, 7, 2)$.

Input	Output	Description
`7 3`		n and m are given.
	`? 1 7`	Ask the position of the minimum in the range $p[1, 7]$.
`1`		The judge responds with $i = 1$.
	`? 4 4`	Ask the position of the minimum in the range $p[4, 4]$.
`4`		The judge responds with $i = 4$.
	`? 2 7`	Ask the position of the minimum in the range $p[2, 7]$.
`7`		The judge responds with $i = 7$.
	`? 2 6`	Ask the position of the minimum in the range $p[2, 6]$.
`2`		The judge responds with $i = 2$.
	`! 2`	Print $k = 2$ as the answer.

For $k = 2$, we have $a_k = m$. Thus, if the program immediately quits here, this case will be judged as correctly solved.

Here is another example with $n = 4$, $m = 2$, and $p = (2, 1, 4, 3)$.

Input	**Output**	**Description**
`4 2`		n and m are given.
	`? 1 4`	Ask the position of the minimum in the range $p[1, 4]$.
`2`		The judge responds with $i = 2$.
	`! -1`	Print `-1` as the answer since it is impossible to find the index k such that $a_k = m$.

E. Different Adjacent

Limits: 2 sec., 512 MiB

You are given a sequence a of n integers.
Construct a sequence b of n integers satisfying the following conditions.

- $b_i \geq a_i$ for all $1 \leq i \leq n$.
- Adjacent elements of b are distinct.
- $\max_{1 \leq i \leq n}(b_i - a_i)$ is minimum possible.

Input

The first line contains an integer n – the number of elements in sequence a.
The second line contains n integers a_i – the elements of sequence a.

Output

Print n integers b_i – elements of sequence b.

Constraints

$1 \leq n \leq 2 \cdot 10^5$,
$1 \leq a_i \leq 10^9$.

Samples

Input (*stdin*)	Output (*stdout*)
4 4 7 4 7	4 7 4 7
7 4 4 4 4 4 4 4	5 4 5 4 5 4 5

Notes

In the first example, it is possible to make $b = a$. In this case, $\max_{1 \leq i \leq n}(b_i - a_i) = 0$.
In the second example, $\max_{1 \leq i \leq n}(b_i - a_i) = 1$.

F. Minimize the Maximum Distance

Limits: 3 sec., 512 MiB

This problem is different from the problem "Maximize the Minimum Distance".

There are $n \cdot m$ points on a two-dimensional plane, located in n rows and m columns.

The distance between the i-th and the $(i+1)$-th row is a_i.

The distance between the i-th and the $(i+1)$-th column is b_i.

You want to paint all the points in some colors. The number of colors that you can use is unlimited. However, the number of points for each color you use must be two or three.

Your goal is to minimize the maximum Euclidean distance between points of the same color. You only need to find the value of the sought distance and do not need to find the actual colors of the points.

Input

The first line contains two integers n and m – the number of rows and columns, respectively.

The second line contains $n-1$ integers a_i – distances between the i-th and the $(i+1)$-th row.

The third line contains $m-1$ integers b_i – distances between the i-th and the $(i+1)$-th column.

Output

In one line print a real number – the minimum possible maximum distance between points of the same color.

Your output will be considered correct if the absolute or relative error from the true answer is at most 10^{-7}.

Constraints

$2 \le n, m \le 3 \cdot 10^5$,

$1 \le a_i, b_i \le 10^9$.

Samples

Input (*stdin*)	Output (*stdout*)
2 3 1 4 1	1.0000000000
3 3 1 1 1 1	1.4142135624

Notes

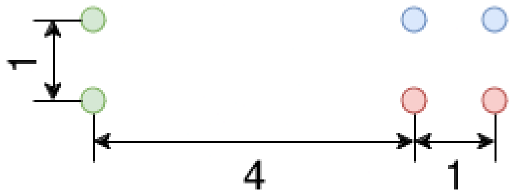


Figure 3: In the first example, we have two rows and three columns of points. We paint them in three colors, with two points for each color. Euclidean distance between two points of each color equals 1.

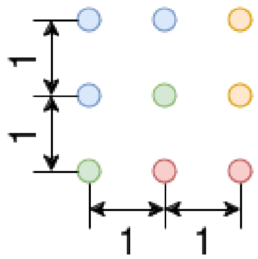


Figure 4: In the second example, we paint points in four colors: three in cyan-blue color and two points in each of green, red, and brown. Two farther cyan-blue points are at the distance $\sqrt{2}$. Also, two green points are at the distance $\sqrt{2}$. The answer for this example is $\sqrt{2}$.

G. Maximize the Minimum Distance

Limits: 2 sec., 512 MiB

This problem is different from the problem "Minimize the Maximum Distance".

There are $n \cdot m$ points on a two-dimensional plane located in n rows and m columns.

Distances between adjacent rows and columns are all equal to 1.

You want to paint all the points in some colors. We denote all colors you can use with integers $1, 2, \ldots, 10^5$. The number of points for each color you use must be two or three.

Your goal is to maximize the minimum Euclidean distance between points of the same color. You need not only to find the value of the sought distance but also to find the actual colors of the points.

Input

The single line contains two integers n and m – the number of rows and columns, respectively.

Output

In the first line, print an integer number – the square of the maximum possible minimum distance between points of the same color.

In the following n lines, print m integers c_{ij} that denote the colors of the points. These numbers must satisfy $1 \leq c_{ij} \leq 10^5$. Each of these numbers must occur exactly two or three times.

If there are multiple answers, any of them will be accepted.

Constraints

$1 \leq n, m$,
$2 \leq n \cdot m \leq 10^5$.

Samples

Input (*stdin*)	Output (*stdout*)
2 3	2 4 7 4 7 4 7
4 3	5 1 2 3 3 4 5 6 1 2 4 5 6
1 5	4 4 7 4 7 4

Notes

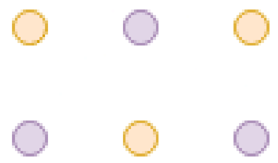


Figure 5: In the first example, the sought distance is $\sqrt{2}$.

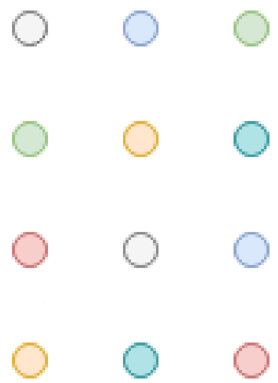


Figure 6: In the second example, the sought distance is $\sqrt{5}$.

Figure 7: In the third example, the sought distance is 2.

H. Laminaria

Limits: 5 sec., 1024 MiB

You are given two trees T_1 and T_2 of n vertices. In both trees, the vertices are numbered 1 through n. Also, in both trees, the i-th vertex has color i for all $1 \le i \le n$.

You want to choose a vertex u from T_1, a vertex v from T_2, and connect them with an edge. This way, you will obtain a new tree T of $2n$ vertices with two vertices of each color from 1 to n.

Then, you will perform the following operation of marking vertices of T. Initially, all vertices are not marked.

- Choose two vertices of the same color i that are not marked, and all internal vertices on the path between them are marked. Mark both vertices of color i.

We call a tree T a *laminaria* if it is possible to mark all its vertices applying the above operation any number of times.

In how many ways can you connect T_1 and T_2 to obtain a laminaria?

Input

The first line contains an integer n – the number of vertices in both given trees.

The following $n-1$ lines contain two integers u_i, v_i each – the ends of the i-th edge in T_1.

The following $n-1$ lines contain the edges of T_2 in the same format.

Output

Print a single integer – the number of ways to connect T_1 and T_2 to obtain a laminaria.

Constraints

$1 \le n \le 2 \cdot 10^5$,
$1 \le u_i, v_i \le n$.

Samples

Input (*stdin*)	Output (*stdout*)
5 3 1 2 4 2 5 1 2 1 2 2 3 3 5 5 4	2
4 1 2 2 3 3 4 1 2 2 3 3 4	4

Notes

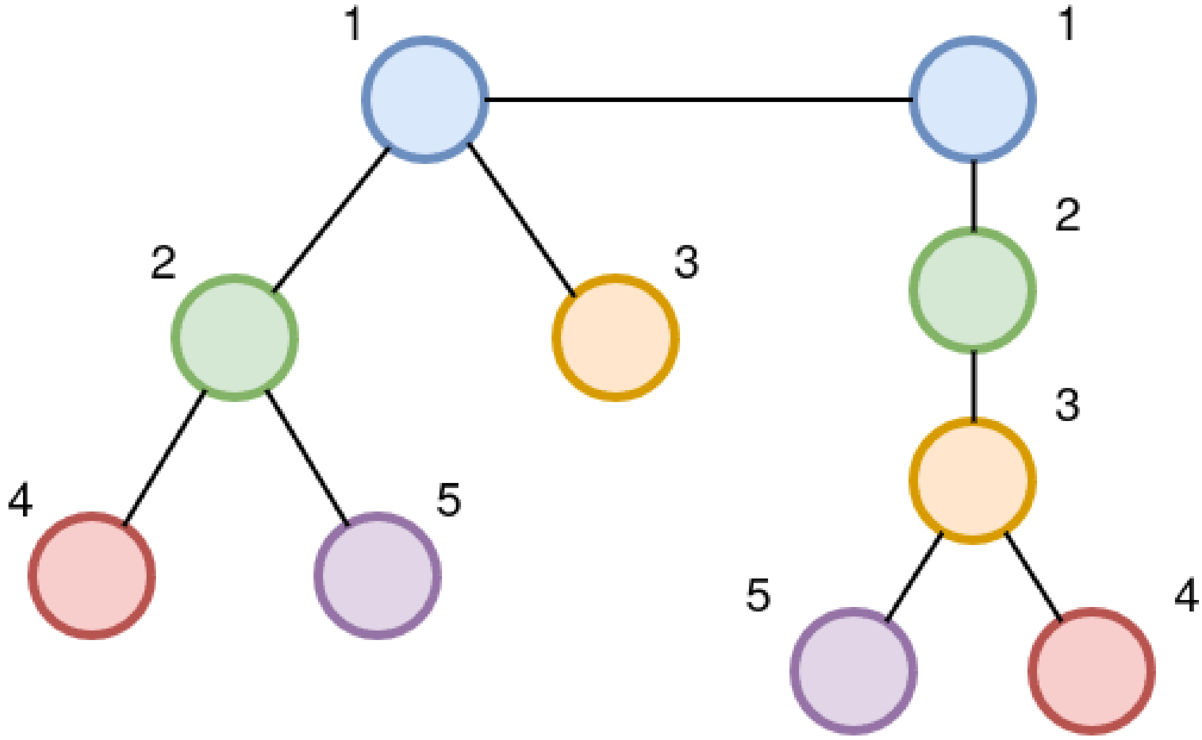


Figure 8: In the first example, we can connect both vertices of color 1 to obtain a laminaria. If we choose the colors in order of $1, 2, 3, 5, 4$, we will mark all vertices. It is not the only way to obtain a laminaria, however. Another way is to connect both vertices of color 2. The total number of ways to obtain a laminaria is 2.

I. Minimum Divisible Sequence

Limits: 2 sec., 512 MiB

You are given an integer n.
Construct an integer sequence a of length m satisfying the following conditions.

- $1 \le a_i \le n$ for all $1 \le i \le m$.
- For each $1 \le k \le n$, there exists $1 \le i \le m$ such that a_i is a multiple of k.
- The length m is minimum possible.

Input

The single line contains an integer n.

Output

In the first line print an integer m – the length of the sequence a.
In the second line print m integers a_i – the elements of the sequence.
If there are multiple answers, any of them will be accepted.

Constraints

$1 \le n \le 10^5$.

Samples

Input (*stdin*)	Output (*stdout*)
7	4 4 7 5 6

Notes

In the example, one of the possible sequences is $a = (4, 7, 5, 6)$.

- $a_1 = 4$ is a multiple of 1.
- $a_1 = 4$ is a multiple of 2.
- $a_4 = 6$ is a multiple of 3.
- $a_1 = 4$ is a multiple of 4.
- $a_3 = 5$ is a multiple of 5.
- $a_4 = 6$ is a multiple of 6.
- $a_2 = 7$ is a multiple of 7.

It is impossible to construct a sequence satisfying the conditions of length less than four.

J. Permutations of Two Arrays

Limits: 2 sec., 512 MiB

You are given two sequences: a of length n and b of length m.

Let $k = \min(n, m)$.

You want to choose a permutation c of the sequence a and a permutation d of the sequence b to maximize the following score:

$$\sum_{i=1}^{k} |c_i - d_i|.$$

Input

The first line contains two integers n and m – the lengths of the sequences a and b, respectively.

The second line contains n integers a_i.

The third line contains m integers b_i.

Output

In the single line print an integer – the maximum score.

Constraints

$1 \le n, m \le 10^5$,
$0 \le a_i, b_i \le 10^9$.

Samples

Input (*stdin*)	Output (*stdout*)

`4 4`
`4 7 7 4`
`44 47 4 7`
	`86`

`4 7`
`1 2 3 4`
`10 20 30 40 50 60 70`
	`210`

Notes

In the first example, we have $a = (4, 7, 7, 4), b = (44, 47, 4, 7)$. When we choose $c = (7, 4, 4, 7), d = (7, 47, 44, 4)$, the score will be $|c_1 - d_1| + |c_2 - d_2| + |c_3 - d_3| + |c_4 - d_4| = |7-7| + |4-47| + |4-44| + |7-4| = 0 + 43 + 40 + 3 = 86$. It is the maximum score.

K. Values in a Rooted Tree

Limits: 3 sec., 512 MiB

We have a rooted tree with n vertices. The vertices are numbered 1 through n, and the root is vertex 1. The i-th edge connects vertices u_i and v_i. Vertex v has an integer a_v written on it.

For each vertex v, find the maximum m such that there exists x, such that all values $x, x+1, \ldots, x+m-1$ are present on the path between the root and vertex v.

Input

The first line contains an integer n – the number of vertices in the tree.

The second line contains n integers a_v written on the vertices of the tree.

The following $n-1$ lines contain two integers u_i, v_i each – the ends of the i-th edge.

Output

In a single line, print n integers – the maximum m for each $1 \leq v \leq n$.

Constraints

$1 \leq n \leq 3 \cdot 10^5$,
$0 \leq a_v \leq 10^9$,
$1 \leq u_i, v_i \leq n$.

Samples

Input (*stdin*)	Output (*stdout*)
5 2 4 0 3 5 1 2 1 3 2 4 2 5	1 1 1 3 2
7 1 3 2 2 0 11 10 4 5 5 1 2 5 2 3 1 6 7 6	1 2 4 3 2 1 2

Notes

In the first example, the answers for all vertices are the following.

- Vertex 1. Since the root is vertex 1, thus the path between the root and vertex 1 contains the only vertex. In this case, $m = 1$.

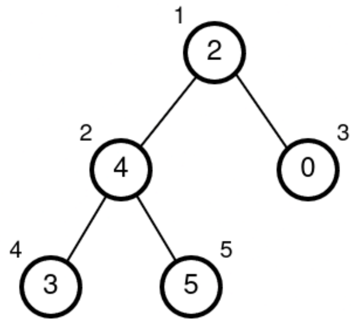


Figure 9: The first example

- Vertex 2. The path between the root and the vertex 2 contains two vertices 1 and 2 with values 2 and 4, respectively. Again, $m = 1$.
- Vertex 3. The path between the root and the vertex 3 contains two vertices 1 and 3 with values 2 and 0, respectively. For vertex 3, the answer is $m = 1$.
- Vertex 4. The path between the root and the vertex 4 contains three vertices 1, 2 and 4 with values 2, 4 and 3 respectively. Here the answer is $m = 3$, because for $x = 2$, the values $x = 2$, $x + 1 = 3$ and $x + 2 = 4$ are present on the path.
- Vertex 5. The path between the root and the vertex 5 contains three vertices 1, 2 and 5 with values 2, 4 and 5 respectively. Here, the answer is $m = 2$ because, for $x = 4$, the values $x = 4$ and $x + 1 = 5$ are present on the path.

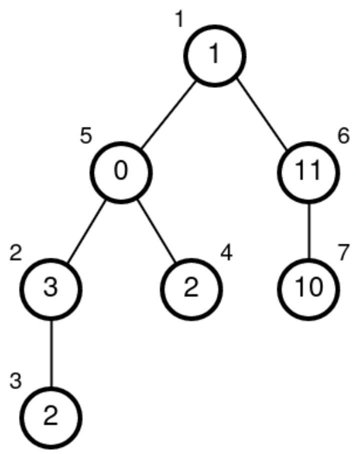
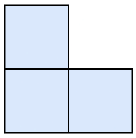


Figure 10: The second example

L. L-trominos

Limits: 2 sec., 512 MiB

We have a rectangular grid of 2 rows and n columns.
Also, you have an infinite supply of *L-trominos*:



You want to tile the grid with L-trominos to satisfy the following conditions.

- When placing each tromino, rotation, and reflection are allowed.
- Each tile must align with cells.
- Each cell in the grid must be covered by at most one tromino.
- No part of each tromino may be outside the grid.

Find the minimum number of uncovered cells in the grid.

Input

The single line contains an integer n – the number of columns in the grid.

Output

Print a single integer – the minimum number of uncovered cells in the grid.

Constraints

$1 \le n \le 10^6$.

Samples

Input (*stdin*)	Output (*stdout*)
3	0
4	2
47	1

Notes

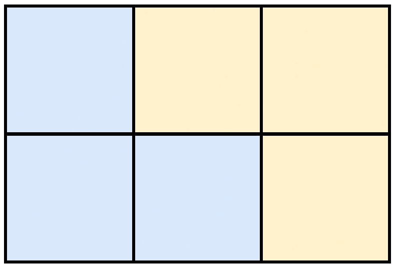


Figure 11: In the first example, it is possible to cover all cells.

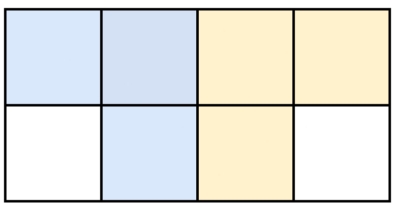


Figure 12: In the second example, the minimum number of uncovered cells is two.