Meetings

The Society for Saving the World has called their N members to an emergency congress to finally agree on a plan for saving the world. To reach a common decision in any meeting at the congress, the meeting participants proceed as follows:

1. Each of them has a proposal and takes P minutes to present it to the others.
2. After all participants have made their presentations, they vote for the best proposal, which takes V minutes.

For example, if each proposal takes one minute ($P = 1$) and voting also takes one minute ($V = 1$), a meeting with 100 participants would reach a decision in 101 minutes.

To speed up the overall decision-making process, the participants of the congress have decided to split into groups and work in parallel. Each group selects the best proposal among themselves using the procedure described above. Then the representatives of the groups meet and pick the final plan among the proposals voted best in each group.

For example, if the 100 participants would split into two groups of 40 and 60, respectively, the process could work as follows (again, $P = V = 1$):

- the larger group takes 61 minutes to select their best proposal;
- the smaller group takes 41 minutes to do the same and then has to wait for the larger group to finish;
- then the two representatives of the groups meet and spend 2 minutes presenting to each other and 1 minute to vote.

The total time spent is thus $61 + 2 + 1 = 64$ minutes.

But the groups might further divide themselves into subgroups and sometimes it could be useful to split into more than two groups. As a special case, a subgroup of 1 member can decide in no time, as there is no need to present one's own proposal to oneself.

Write a program that, given presentation and voting times P and V, computes the minimal time needed for the N participants of the congress to reach a common decision, assuming they organize their meetings and groups optimally.

Input

The first and only line of input contains the three integers N, P, and V: N is the number of participants of the congress, P is the presentation time (in minutes), and V is the voting time (in minutes).

Output

The first and only line of output should contain the integer M, the minimal number of minutes needed for the congress to reach a decision.

Constraints

$1 \leq N \leq 10^{15}$
$1 \leq P, V \leq 1\,000$
In test cases worth 70 points, $1 \leq N \leq 50\,000$.
In test cases worth 40 points, $1 \leq N \leq 5\,000$.

Examples

Input	Output
`9 1 1`	`8`
`6 1 2`	`8`
`6 2 1`	`12`

In the first example, the participants can be divided into 3 groups of 3 members each. Then each group needs 4 minutes and the 3 representatives need another 4 minutes for their final meeting.

Plagiarism

The participants of the World Programming Competition submitted N solution files $f_1, \ldots, f_N$ to the grading system. Before accepting the results as final, the jury would like to rule out any possibility of plagiarism. They have a program that takes two files and compares them to decide if they are too similar to each other.

However, the number of files is rather big and it would take too much time to compare all pairs. On the other hand, many pairs could be quickly eliminated based on the fact that the file sizes are too different.

More precisely, the jury decided to fully skip comparing every pair where the size of the smaller file is less than 90% of the size of the larger one. So, the comparison program has to examine only those distinct pairs of files (f_i, f_j) where $i \neq j$, size$(f_i) \leq$ size(f_j) and size$(f_i) \geq 0.9 \cdot$ size(f_j).

Write a program that computes the number of pairs of files that will have to be examined.

Input

The first line of input contains the integer N, the number of solution files submitted. The second line contains N integers size$(f_1), \ldots,$ size(f_N), each showing the size of one file.

Output

The first and only line of output must contain one integer, the number of pairs of files that will have to be examined.

Constraints

$1 \leq N \leq 100\,000$
$1 \leq \text{size}(f_i) \leq 100\,000\,000$
In test cases worth 50 points, $1 \leq N \leq 2\,000$.

Examples

Input	Output
`2` `2 1`	`0`
`5` `1 1 1 1 1`	`10`

In the second example, each file has to be compared to each other (but each pair only once, not twice, of course).

Polygon

A simple polygon with N vertices is drawn on an infinite rectangular grid. For such a polygon, only neighboring edges touch at their common vertex; no other of its edges intersect or touch. All vertices of the polygon lie on grid points, i.e., vertices have integer coordinates.

Your task is to find the total length of grid line segments which lie strictly inside the given polygon (these line segments are highlighted in the drawings below).

Input

The first line of input contains a single integer N, the number of vertices of the polygon. Each of the following N lines contains two integers x and y, the coordinates of one vertex. The vertices are given either in clockwise or counterclockwise order. All vertices are distinct, but more than two consecutive vertices may lie on a line.

Output

The only line of output must contain a decimal number: the total length of grid line segments which lie strictly inside the given polygon.

Constraints

$3 \leq N \leq 100\,000$
$-500\,000\,000 \leq x, y \leq 500\,000\,000$
In test cases worth 50 points, all polygon edges lie on grid lines.

Grading

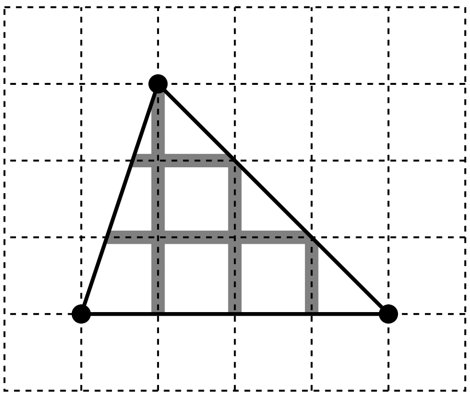
Your output is graded correct if it is close enough to the expected output.

To be more precise: Let your output number be L, and let the expected result be R. Then, at least one of the two following conditions must hold:

- $|L - R| \leq R \cdot 10^{-6}$ *(relative precision)*
- $|L - R| \leq 10^{-6}$ *(absolute precision)*

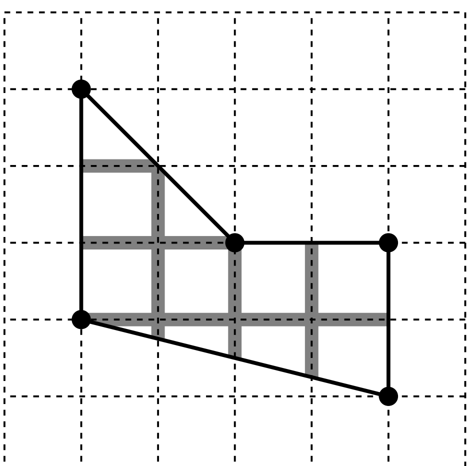
Examples

Input	Output
3 5 1 2 4 1 1	10.0



The length of horizontal lines is $4/3 + 8/3 = 4$. The length of vertical lines is $3 + 2 + 1 = 6$. The total length is $4 + 6 = 10$.

Input	Output
5 0 0 -2 2 -2 -1 2 -2 2 0	12.5



The length of horizontal lines is $1 + 2 + 4 = 7$. The length of vertical lines is $9/4 + 3/2 + 7/4 = 5.5$. The total length is $7 + 5.5 = 12.5$.

Tree Mirroring

Let T be a rooted tree (a connected undirected acylic graph), and let S be a perfect copy of T. Construct a new graph by taking the union of T and S, and merging the corresponding leaf nodes (but never the root). We call such a graph a *tree-mirrored graph*.

Write a program that determines if an arbitrary undirected connected graph is a tree-mirrored graph.

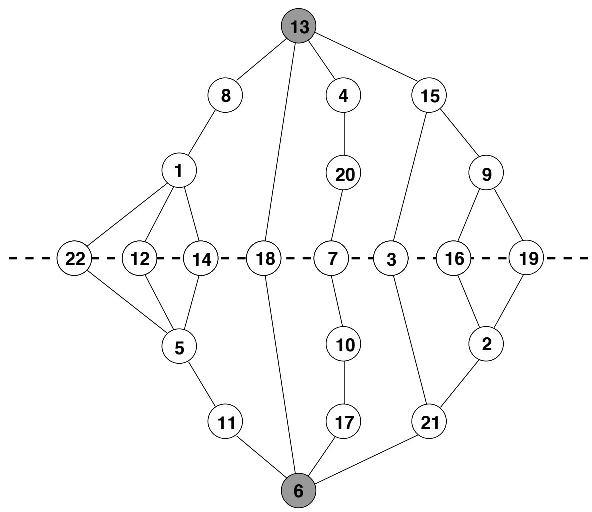


Figure 1: An example of a tree-mirrored graph. The figure corresponds to the third example test case.

Input

The first line of input contains two integers N and M, the number of vertices and edges of a graph G. The vertices in G are labeled from 1 to N. The following M lines describe the edges. Each such line contains two integers x and y ($x \neq y$; $1 \leq x, y \leq N$), describing one edge. There will be at most one edge between any pair of vertices.

Output

The first and only line of output should contain the string `YES` if the graph G is a tree-mirrored graph, and `NO` otherwise.

Constraints

$3 \leq N, M \leq 100\,000$
In test cases worth 60 points, $3 \leq N, M \leq 3\,500$. In test cases worth 30 points, $3 \leq N, M \leq 300$.

Examples

Input	Output
7 7 1 2 2 3 3 4 4 5 5 6 6 7 7 1	NO
6 6 1 2 2 3 2 4 3 5 4 5 5 6	YES
22 28 13 8 8 1 1 22 1 12 1 14 13 18 13 4 4 20 20 7 13 15 15 3 15 9 9 16 9 19 22 5 12 5 14 5 5 11 11 6 18 6 7 10 10 17 17 6 3 21 21 6 16 2 19 2 2 21	YES

The last example input corresponds to the graph in the figure.