A. Birthday

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

Ivan is collecting coins. There are only N different collectible coins, Ivan has K of them. He will be celebrating his birthday soon, so all his M freinds decided to gift him coins. They all agreed to three terms:

- Everyone must gift as many coins as others.
- All coins given to Ivan must be different.
- Not less than L coins from gifts altogether, must be new in Ivan's collection.

But his friends don't know which coins have Ivan already got in his collection. They don't want to spend money so they want to buy minimum quantity of coins, that satisfy all terms, irrespective of the Ivan's collection. Help them to find this minimum number of coins or define it's not possible to meet all the terms.

Input

The only line of input contains 4 integers N, M, K, L ($1 \le K \le N \le 10^{18}$; $1 \le M, \ L \le 10^{18}$) — quantity of different coins, number of Ivan's friends, size of Ivan's collection and quantity of coins, that must be new in Ivan's collection.

Output

Print one number — minimal number of coins one friend can gift to satisfy all the conditions. If it is impossible to satisfy all three conditions print "-1" (without quotes).

Examples

input
```
20 15 2 3
```
output
```
1
```

input
```
10 11 2 4
```
output
```
-1
```

Note

In the first test, one coin from each friend is enough, as he will be presented with 15 different coins and 13 of them will definitely be new.

In the second test, Ivan has 11 friends, but there are only 10 different coins. So all friends can't present him different coins.

B. LCM

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

Ivan has number b. He is sorting through the numbers a from 1 to 10^{18}, and for every a writes $\frac{[a, \ b]}{a}$ on blackboard. Here $[a, \ b]$ stands for least common multiple of a and b. Ivan is very lazy, that's why this task bored him soon. But he is interested in how many different numbers he would write on the board if he would finish the task. Help him to find the quantity of different numbers he would write on the board.

Input

The only line contains one integer — b $(1 \le b \le 10^{10})$.

Output

Print one number — answer for the problem.

Examples

input
```
1
```

output

```
1
```

input

```
2
```

output

```
2
```

Note

In the first example $[a,\ 1] = a$, therefore $\frac{[a,\ b]}{a}$ is always equal to 1.

In the second example $[a,\ 2]$ can be equal to a or $2 \cdot a$ depending on parity of a. $\frac{[a,\ b]}{a}$ can be equal to 1 and 2.

C. Colored Rooks

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

Ivan is a novice painter. He has n dyes of different colors. He also knows exactly m pairs of colors which harmonize with each other.

Ivan also enjoy playing chess. He has 5000 rooks. He wants to take k rooks, paint each of them in one of n colors and then place this k rooks on a chessboard of size $10^9 \times 10^9$.

Let's call the set of rooks on the board *connected* if from any rook we can get to any other rook in this set moving only through cells with rooks from this set. Assume that rooks can jump over other rooks, in other words a rook can go to any cell which shares vertical and to any cell which shares horizontal.

Ivan wants his arrangement of rooks to have following properties:

- For any color there is a rook of this color on a board;
- For any color the set of rooks of this color is connected;
- For any two different colors a b union of set of rooks of color a and set of rooks of color b is connected if and only if this two colors harmonize with each other.

Please help Ivan find such an arrangement.

Input

The first line of input contains 2 integers n, m ($1 \le n \le 100$, $0 \le m \le min(1000,\ \frac{n(n-1)}{2})$) — number of colors and number of pairs of colors which harmonize with each other.

In next m lines pairs of colors which harmonize with each other are listed. Colors are numbered from 1 to n. It is guaranteed that no pair occurs twice in this list.

Output

Print n blocks, i-th of them describes rooks of i-th color.

In the first line of block print one number a_i ($1 \le a_i \le 5000$) — number of rooks of color i. In each of next a_i lines print two integers x and y ($1 \le x,\ y \le 10^9$) — coordinates of the next rook.

All rooks must be on different cells.

Total number of rooks must not exceed 5000.

It is guaranteed that the solution exists.

Examples

input

```
3 2
1 2
2 3
```

output

```
2
3 4
1 4
4
1 2
2 2
2 4
5 4
1
5 1
```

input

```
3 3
1 2
2 3
3 1
```

output

```
1
1 1
1
1 2
1
1 3
```

input

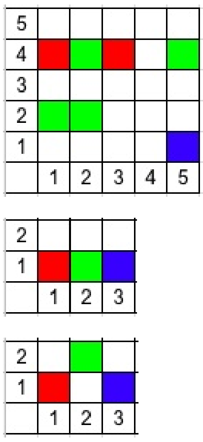
```
3 1
1 3
```

output

```
1
1 1
1
2 2
1
3 1
```

Note

Rooks arrangements for all three examples (red is color 1, green is color 2 and blue is color 3).



D. Array Without Local Maximums

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Ivan unexpectedly saw a present from one of his previous birthdays. It is array of n numbers from 1 to 200. Array is old and some numbers are hard to read. Ivan remembers that for all elements at least one of its neighbours ls not less than it, more formally:

$a_1 \le a_2$,

$a_n \le a_{n-1}$ and

$a_i \le max(a_{i-1},\ a_{i+1})$ for all i from 2 to $n-1$.

Ivan does not remember the array and asks to find the number of ways to restore it. Restored elements also should be integers from 1 to 200. Since the number of ways can be big, print it modulo 998244353.

Input

First line of input contains one integer n ($2 \le n \le 10^5$) — size of the array.

Second line of input contains n integers a_i — elements of array. Either $a_i = -1$ or $1 \le a_i \le 200$. $a_i = -1$ means that i-th element can't be read.

Output

Print number of ways to restore the array modulo 998244353.

Examples

input

```
3
1 -1 2
```

output

```
1
```

input

```
2
-1 -1
```

output

```
200
```

Note

In the first example, only possible value of a_2 is 2.

In the second example, $a_1 = a_2$ so there are 200 different values because all restored elements should be integers between 1 and 200.

E. Multihedgehog

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

Someone give a strange birthday present to Ivan. It is hedgehog — connected undirected graph in which one vertex has degree at least 3 (we will call it center) and all other vertices has degree 1. Ivan thought that hedgehog is too boring and decided to make himself k-multihedgehog.

Let us define k-multihedgehog as follows:

- 1-multihedgehog is hedgehog: it has one vertex of degree at least 3 and some vertices of degree 1.
- For all $k \ge 2$, k-multihedgehog is $(k-1)$-multihedgehog in which the following changes has been made for each vertex v with degree 1: let u be its only neighbor; remove vertex v, create a new hedgehog with center at vertex w and connect vertices u and w with an edge. New hedgehogs can differ from each other and the initial gift.

Thereby k-multihedgehog is a tree. Ivan made k-multihedgehog but he is not sure that he did not make any mistakes. That is why he asked you to check if his tree is indeed k-multihedgehog.

Input

First line of input contains 2 integers n, k ($1 \le n \le 10^5$, $1 \le k \le 10^9$) — number of vertices and hedgehog parameter.

Next $n-1$ lines contains two integers u v ($1 \le u,\ v \le n;\ u \ne v$) — indices of vertices connected by edge.

It is guaranteed that given graph is a tree.

Output

Print "`Yes`" (without quotes), if given graph is k-multihedgehog, and "`No`" (without quotes) otherwise.

Examples

input

```
14 2
1 4
2 4
3 4
4 13
10 5
11 5
12 5
14 5
5 13
6 7
8 6
13 6
9 6
```

output

```
Yes
```

input

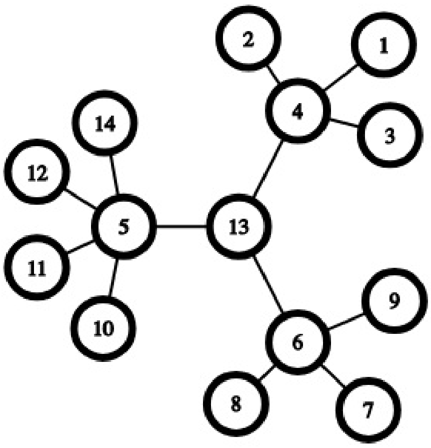
```
3 1
1 3
2 3
```

output

```
No
```

Note

2-multihedgehog from the first example looks like this:



Its center is vertex 13. Hedgehogs created on last step are: [4 (center), 1, 2, 3], [6 (center), 7, 8, 9], [5 (center), 10, 11, 12, 13].

Tree from second example is not a hedgehog because degree of center should be at least 3.

F. Knights

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

Ivan places knights on infinite chessboard. Initially there are n knights. If there is free cell which is under attack of at least 4 knights then he places new knight in this cell. Ivan repeats this until there are no such free cells. One can prove that this process is finite. One can also prove that position in the end does not depend on the order in which new knights are placed.

Ivan asked you to find initial placement of exactly n knights such that in the end there will be at least $\lfloor \frac{n^2}{10} \rfloor$ knights.

Input

The only line of input contains one integer n ($1 \le n \le 10^3$) — number of knights in the initial placement.

Output

Print n lines. Each line should contain 2 numbers x_i and y_i ($-10^9 \le x_i, \ y_i \le 10^9$) — coordinates of i-th knight. For all $i \neq j$, $(x_i, \ y_i) \neq (x_j, \ y_j)$ should hold. In other words, all knights should be in different cells.

It is guaranteed that the solution exists.

Examples

input

```
4
```

output

```
1 1
3 1
1 5
4 4
```

input

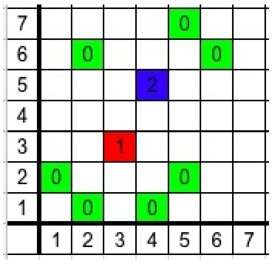
```
7
```

output

```
2 1
1 2
4 1
5 2
2 6
5 7
6 6
```

Note

Let's look at second example:



Green zeroes are initial knights. Cell $(3,\ 3)$ is under attack of 4 knights in cells $(1,\ 2)$, $(2,\ 1)$, $(4,\ 1)$ and $(5,\ 2)$, therefore Ivan will place a knight in this cell. Cell $(4,\ 5)$ is initially attacked by only 3 knights in cells $(2,\ 6)$, $(5,\ 7)$ and $(6,\ 6)$. But new knight in cell $(3,\ 3)$ also attacks cell $(4,\ 5)$, now it is attacked by 4 knights and Ivan will place another knight in this cell. There are no more free cells which are attacked by 4 or more knights, so the process stops. There are 9 knights in the end, which is not less than $\lfloor \frac{7^{2}}{10} \rfloor = 4$.