2021 東大山豬盃線上程式競賽

2021 NTTU Online Programming Contest

- **System URL: http://algotutor.nttu.edu.tw/domjudge**
- **Problems:** There are 7 problems in the online programming contest, 17 pages including the cover in total.
- **Program Input:** Input to the program are through standard input. Program input may contain one or more test cases. Test cases may be separated by any delimiter as specified in the problem statements
- **Program Output:** All outputs should be directed to the standard output (`screen output`).
- **Time Limits:** Judges will run each submitted programs with certain time limits (given in the table below).

Problem Information

Problem	Runtime Limit
Problem A	0.2 seconds
Problem B	1 second
Problem C	0.5 seconds
Problem D	1 second
Problem E	0.2 seconds
Problem F	0.3 seconds
Problem G	0.1 seconds

Problem A

Time limit: 0.2 seconds

Consider the following function f defined for any natural number n:

$f(n)$ is the number obtained by summing up the squares of the digits of n in decimal (or base-ten).

If $n = 19$, for example, then $f(19) = 82$ because $1^2 + 9^2 = 82$.

Repeatedly applying this function f, some natural numbers eventually become 1. Such numbers are called happy numbers. For example, 19 is a happy number, because repeatedly applying function f to 19 results in:

$f(19) = 1^2 + 9^2 = 82$
$f(82) = 8^2 + 2^2 = 68$
$f(68) = 6^2 + 8^2 = 100$
$f(100) = 1^2 + 0^2 + 0^2 = 1$

However, not all natural numbers are happy. You could try 5 and you will see that 5 is not a happy number. If n is not a happy number, it has been proved by mathematicians that repeatedly applying function f to n reaches the following cycle:

$$4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4.$$

Write a program that decides if a given natural number n is a happy number or not.

Input Format

Your program is to read from standard input. The input consists of a single line that contains an integer, n ($1 \le n \le 1{,}000{,}000{,}000$).

Output Format

Your program is to write to standard output. Print exactly one line. If the given number n is a happy number, print out `HAPPY`; otherwise, print out `UNHAPPY`.

Sample Input 1

```
19
```

Sample Output 1

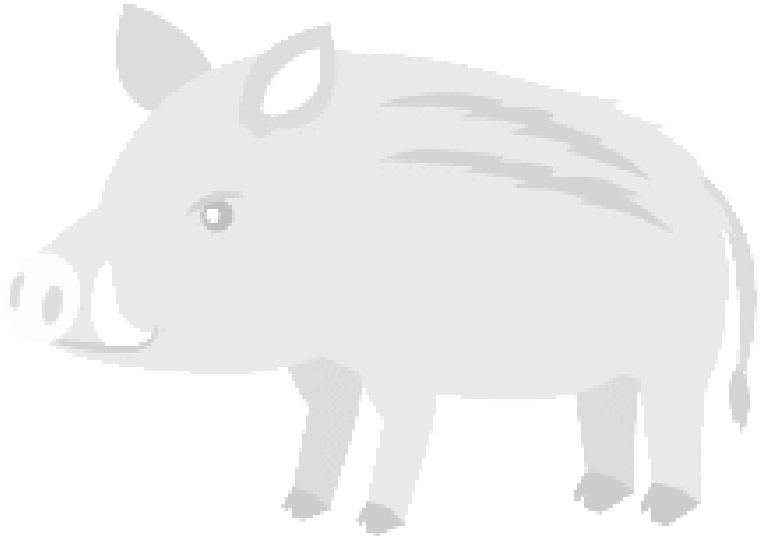
```
HAPPY
```

Sample Input 2

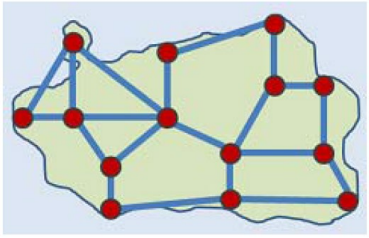
```
5
```

Sample Output 2

```
UNHAPPY
```



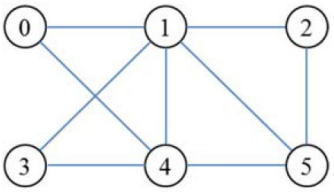
Problem B
Time limit: 1 second



The ICPC-World is the most popular RPG game for ACM-ICPC contestants, whose objective is to conquer the world. A map of the game consists of several cities. There is at most one road between a pair of cities. Every road is bidirectional. If there is a road connecting two cities, they are called neighbors. Each city has one or more neighbors and all cities are connected by one or more roads. A player of the game can start at any city of the world. After conquering a city that the player stays, the player can proceed to any neighbor city which is the city the player to conquer at the next stage.

Chansu, a mania of the game, enjoys the game in a variety of ways. He always determines a list of cities which he wants to conquer before he starts to play the game. In this time, he wants to choose as many cities as possible under the following conditions: Let $(c_0, c_1, ..., c_{m-1})$ be a list of cities that he will conquer in order. All cities of the list are distinct, i.e., $c_i \neq c_j$ if $i \neq j$, c_i and c_{i+1} are neighbors to each other, and the number of neighbors of c_{i+1} is greater than the number of neighbors of ci for $i = 0, 1, \ldots, m - 2$.

For example, let's consider a map of the game shown in the figure below. There are six cities on the map. The city 0 has two neighbors and the city 1 has five neighbors. The longest list of cities satisfying the above conditions is (2, 5, 4, 1) with 4 cities.



In order to help Chansu, given a map of the game with n cities, write a program to find the maximum number of cities that he can conquer, that is, the length of the longest list of cities satisfying the above conditions.

Input Format

Your program is to read from standard input. The input starts with a line containing two integers, n and m ($1 \leq n \leq 100{,}000$, $n - 1 \leq m \leq 300{,}000$), where n is the number of cities on the game map and m is the number of roads. All cities are numbered from 0 to $n - 1$. In the following m lines, each line contains two integers i and j ($0 \leq i \neq j \leq n - 1$) which represent a road connecting two cities i and j.

Output Format

Your program is to write to standard output. Print exactly one line. The line should contain the maximum number of cities which Chansu can conquer.

Sample Input 1

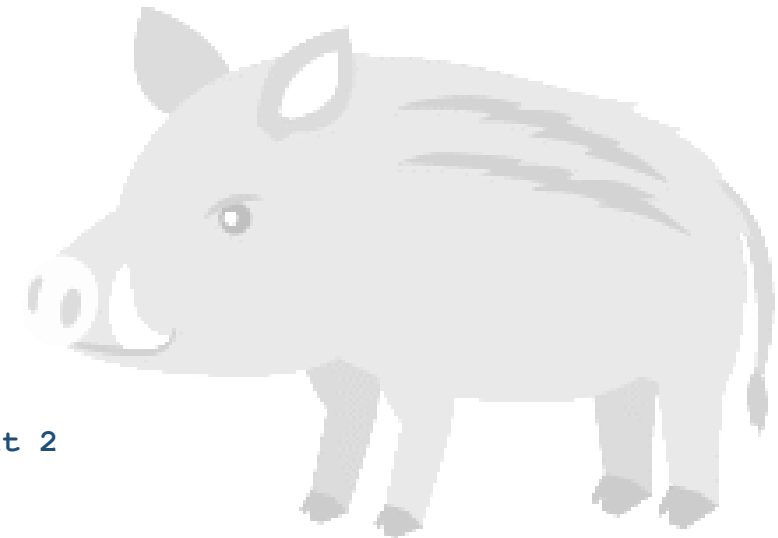
```
6 9
0 1
0 4
1 2
1 3
1 4
1 5
2 5
3 4
4 5
```

Sample Output 1

```
4
```

Sample Input 2

```
12 11
1 2
2 3
3 4
4 5
5 0
6 3
7 4
8 5
9 4
10 5
11 5
```



Sample Output 2

```
5
```

Problem C
Time limit: 0.5 seconds

In Programming Land, there are several pathways called Philosopher's Walks for philosophers to have a rest. A Philosopher's Walk is a pathway in a square-shaped region with plenty of woods. The woods are helpful for philosophers to think, but they planted so densely like a maze that they lost their ways in the maze of woods of a Philosopher's Walk.

Fortunately, the structures of all Philosopher's Walks are similar; the structure of a Philosopher's Walk is designed and constructed according to the same rule in a 2^k meter square. The rule for designing the pathway is to take a right-turn in 90 degrees after every 1-meter step when k is 1, and the bigger one for which the integer k is greater than 1 is built up using four sub-pathways with k – 1 in a fractal style. Figure C.1 shows three Philosopher's Walks for which k is 1, 2, and 3. The Philosopher's Walk W_2 consists of four W_1 structures with the lower-left and the lower-right ones are 90 degree rotated clockwise and counter-clockwise, respectively; the upper ones have the same structure with W_1. The same is true for any W_k for which the integer k is greater than 1. This rule has been devised by a mathematical philosopher David Hilbert (1862-1943), and the resulting pathway is usually called a HILBERT CURVE named after him. He once talked about a space filling method using this kind of curve to fill up a square with 2^k sides, and every Philosophers' Walk is designed according to this method.

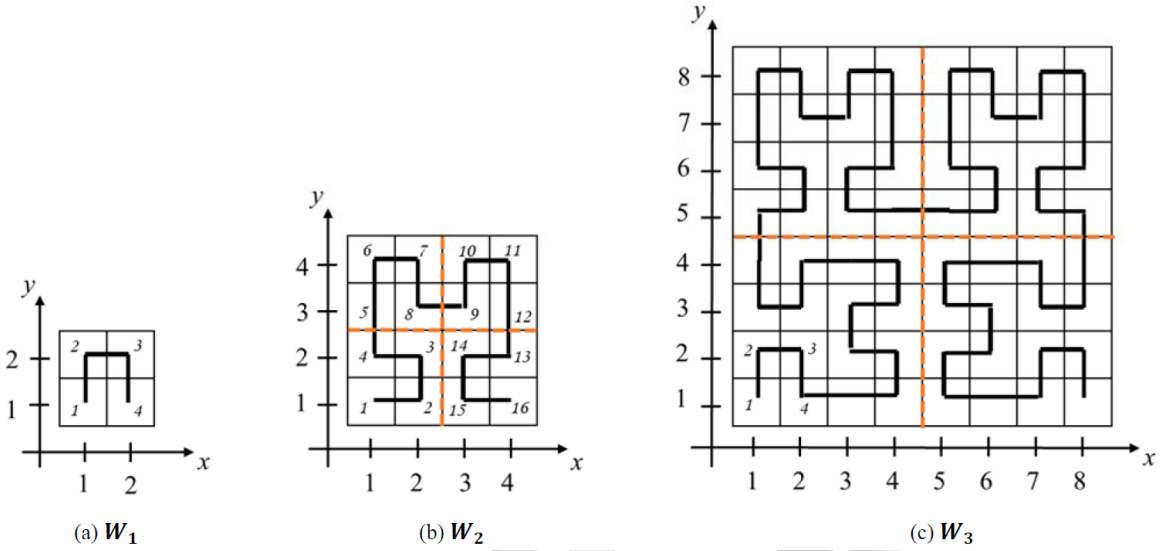


Figure C.1. Three Philosopher's Walks with sizes (a) $2^1 = 2$, (b) $2^2 = 4$, and (c) $2^3 = 8$, respectively.

Jay Lan is in charge of the rescue of the philosophers lost in Philosopher's Walks using a hot air balloon. Fortunately, every lost philosopher can report Jay Lan the number of meter steps he has

taken, and Jay Lan knows the length of a side of the square of the Philosopher's Walk. He has to identify the location of the lost philosopher, the (x, y) coordinates assuming that the Philosopher's Walk is placed in the 1st quadrant of a Cartesian plain with one meter unit length. Assume that the coordinate of the lower-left corner block is (1, 1). The entrance of a Philosopher's Walk is always at (1, 1) and the exit is always $(n, 1)$ where n is the length of a side. Also assume that the philosopher has walked one meter step when he is in the entrance, and that he always go forward to the exit without back steps.

For example, if the number of meter-steps taken by a lost philosopher in the Philosopher's Walk in W_2 in Figure C.1(b) is 10, your program should report (3, 4).

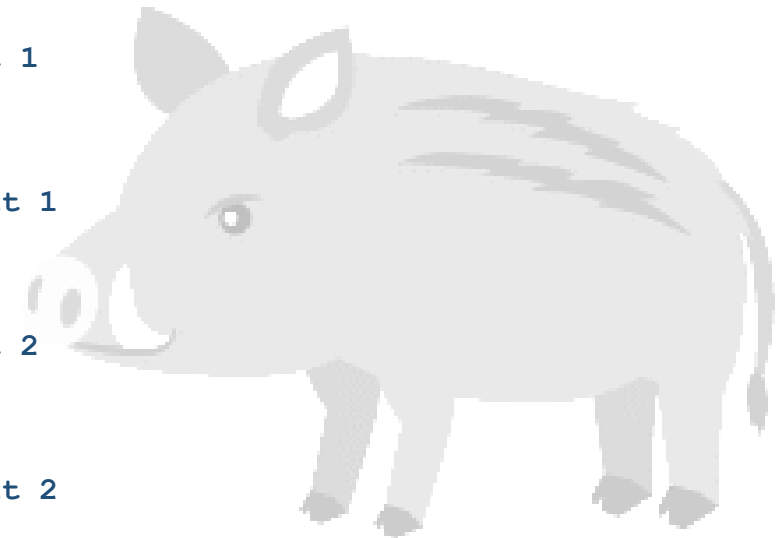
Your mission is to write a program to help Jay Lan by making a program reporting the location of the lost philosopher given the number of meter-steps he has taken and the length of a side of the square of the Philosopher's Walk. Hurry! A philosopher urgently needs your help.

Input Format

Your program is to read from standard input. The input consists of a single line containing two positive integers, n and m, representing the length of a side of the square of the Philosopher's Walk and the number of meter-steps taken by the lost philosopher, respectively, where $n = 2^k$ and $m \le 2^k$ for an integer k satisfying $0 < k \le 15$.

Output Format

Your program is to write to standard output. The single output line should contain two integers, x and y, separated by a space, where (x, y) is the location of the lost philosopher in the given Philosopher's Walk.



Sample Input 1

```
4 10
```

Sample Output 1

```
3 4
```

Sample Input 2

```
8 19
```

Sample Output 2

```
2 6
```

Problem D
Time limit: 1 second

After many years you and your coauthor H. Addaway have finally developed a Theory of Everything that explains everything: Why does time have a direction? How should quantum mechanics be interpreted? What caused the Big Bang? What is love?

An unfortunate fact about physics is that physical theories need to be experimentally tested. In particular, your theory rests on the discovery of so called Barely Audible Particle Clusters (BAPCs). For this purpose you have proposed the development of a Large Eightgon Collider. What remains is to find a suitable location to construct this scientific wonder.

For obvious reasons, the Large Eightgon Collider must consist of eight straight tunnels that together form an underground cycle. Each tunnel is allowed to have a different non-zero length. At each of the eight tunnel connections, a special detector must be built, that also slightly deflects the particles 45 degrees to the left. Each of the eight detectors attracts many researchers, requiring a shaft to the surface to supply them with fresh food and oxygen.

In order to save costs, they will reuse abandoned mine shafts. Given a map of all abandoned mine shafts, your job is to find the number of possible locations to build this miracle. You only consider locations where at least one tunnel runs parallel to the x-axis of the map.

Figure D.1 shows the second sample.

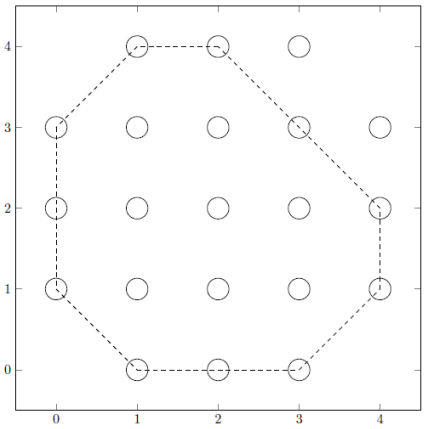


Figure D.1. Visualization of Sample 2 showing one possible location for the Large Eightgon Collider.

Input Format

The input consists of:

- A line with an integer n ($1 \le n \le 5000$), the number of abandoned mine shafts.

- n lines, each with two integers x and y ($-10^8 \le x, y \le 10^8$), the coordinates of the abandoned mine shafts.

Output Format

Output the number of possible locations to build the Large Eightgon Collider.

Sample Input 1

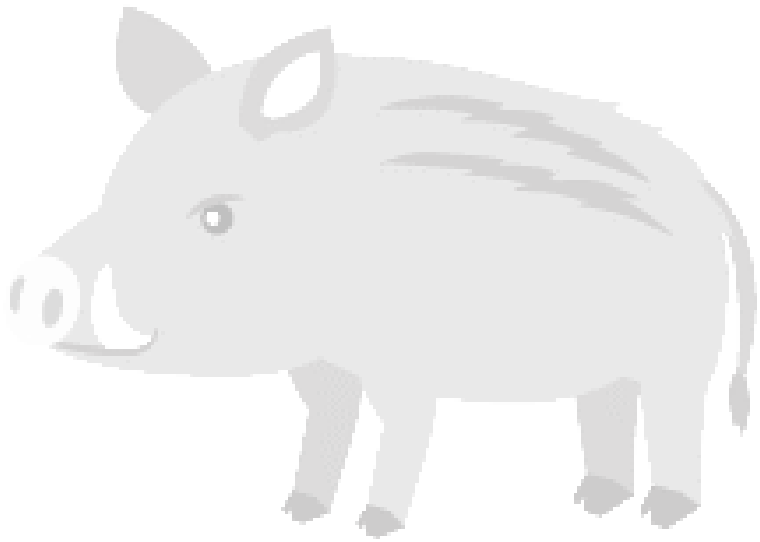
```
8
0 1
1 0
0 2
2 0
3 1
1 3
3 2
2 3
```

Sample Output 1

```
1
```

Sample Input 2

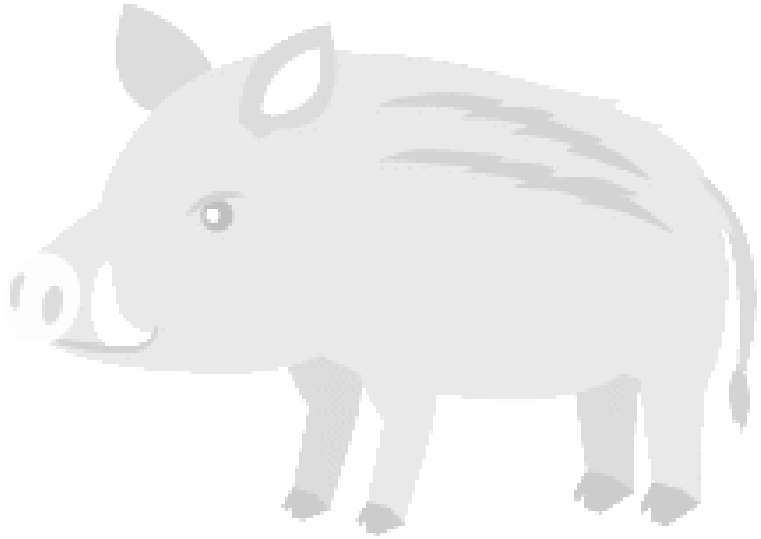
```
21
0 1
0 2
0 3
1 0
1 1
1 2
1 3
1 4
2 0
2 1
2 2
2 3
2 4
3 0
3 1
3 2
```



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

Sample Output 2

```
15
```



Problem E
Time limit: 0.2 seconds

The big day has finally arrived: today you are going to form groups of two in which you will do the end-of-the-year project. When you arrive at school, you learn that the teacher of the other class is sick, and that your teacher, Mr. B.A.P. Cee, will also have to make groups for the other class. Mr. B.A.P. Cee is a smart guy and realizes that he can use these unfortunate circumstances to his advantage.

Ending up with groups of one should be avoided at all cost, so mixing the students of the two classes may avoid this situation. However, while it is easy to pair up two students from the same class, it is more difficult to match up students from different classes. Throughout the years there has been a lot of rivalry between the two groups, and many students dislike students in the other class. Mr. B.A.P. Cee knows which pairs of students will result in a fight and a failed project.

You are given a list of pairs of students who cannot work together. How many disjoint groups of two can Mr. B.A.P. Cee make that will not result in a failed project?

Input Format

The input consists of:

- A line with two integers n ($1 \le n \le 10^5$), the number of students, and m ($0 \le m \le 2 \times 10^5$), the number of pairs of students who cannot work together.
- m lines, each with two distinct integers i and j ($1 \le i, j \le n$, $i \ne j$), giving a pair of students who cannot work together.

Students are identified by the number 1 through n. It is guaranteed that it is possible to split the students into two classes in such a way that all students from the same class get along.

Output Format

Output the number of pairs of students Mr. B.A.P. Cee can make without making any pair of students who cannot work together.

Sample Input 1

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

Sample Output 1

```
1
```

Sample Input 2

```
5 6
1 4
2 4
3 4
1 5
2 5
3 5
```

Sample Output 2

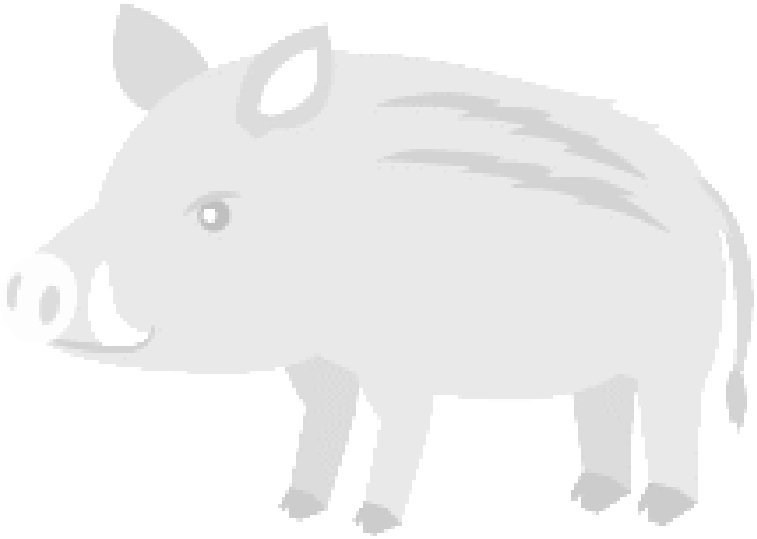
```
2
```

Sample Input 3

```
6 6
1 4
2 5
3 6
1 5
3 5
2 6
```

Sample Output 3

```
3
```



Problem F
Time limit: 0.3 seconds

Bushfires are threatening your habitat! Being a kangaroo, you must inform the other kangaroos in your troop as fast as possible and flee to a safe area.

You are currently standing still, and you will jump to the other kangaroos' locations. You will visit the other kangaroos in a specific order so that all of them have sufficient time to escape. After visiting all other kangaroos you must continue jumping to the safe area, where you should come to a stop.

With each jump you move a (possibly negative) integer distance north and/or east. Because of your limited muscle power, you are only able to accelerate or decelerate at most 1 in each direction each jump. Formally, if jump i moves you $v_{x,i}$ to the north and $v_{y,i}$ to the east, the next jump $i + 1$ must satisfy $|v_{x,i+1} - v_{x,i}| \le 1$ and $|v_{y,i+1} - v_{y,i}| \le 1$.

Find the minimal number of jumps needed to go from your current position to the safe area via all other kangaroos, without leaving the grid. It is very important to come to a full stop at the end, so the last jump must both start and end at the safe area.

The first sample is shown in Figure F.1.

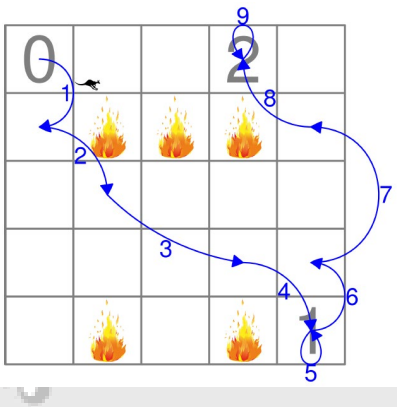


Figure F.1. Visualization of Sample 1 showing one possible way to get to the safe using 9 jumps.

Input Format

The input consists of:

- A line containing three integers r, c ($1 \le r, c \le 50$), the number of rows and columns of the grid, and k ($1 \le k \le 5$), the number of other kangaroos you need to warn.
- r lines each consisting of c characters. A "." indicates an open space and a "#" indicates a bush where you cannot jump.

Your start position is indicated by a "0" and the characters "1" to k indicate the positions of the other kangaroos you need to wain *in this order*.

The positions indicated by $k + 1$ indicates the safe area where you should come to stop.

Output Format

If it is possible to reach the safe area, then output the minimal number of steps needed to reach the safe area. Otherwise, output "impossible".

Sample Input 1

```
5 5 1
0..2.
.###.
.....
.....
.#.#1
```

Sample Output 1

```
9
```

Sample Input 2

```
2 2 2
03
12
```

Sample Output 2

```
4
```

Sample Input 3

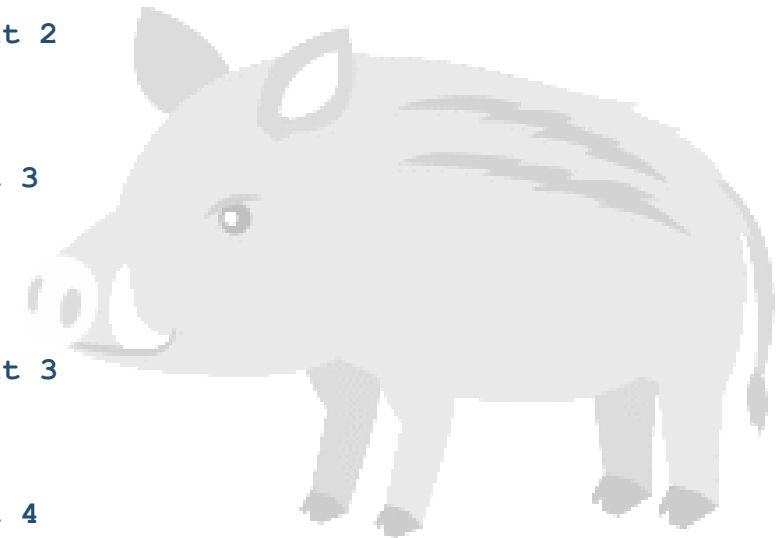
```
1 5 1
.0#21
```

Sample Output 3

```
8
```

Sample Input 4

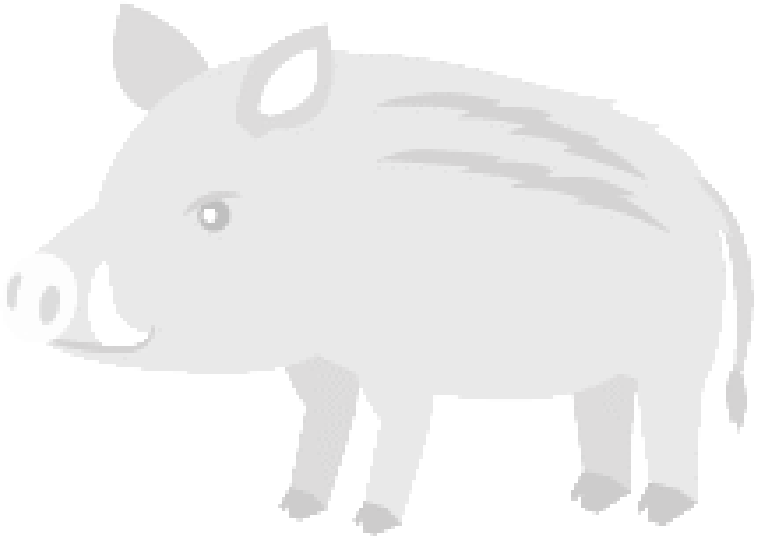
```
3 4 1
#0##
#.#2
```



```
1###
```

Sample Output 4

```
impossible
```



Problem G
Time limit: 0.1 seconds

The fruit harvest has been good this year, which means that your jam-selling company, which produces the price-winning Berry Artisanal and Pure Compote, is shipping out jam left and right! A customer has recently placed a huge order of n jars of jam. To ship these jars, you put them into boxes, each of which can hold up to k jars.

As is always the case with fragile goods, the jars might break in the process of being delivered. You want to avoid the jars bouncing around in their boxes too much, as that significantly increases the chance that they break. To circumvent this, you want to avoid having boxes that are too empty: that would inevitably result in the uncontrolled bouncing around, and subsequently breaking, of the jars. In particular, you want the box with the least number of jars to be as full as possible. In order to estimate the risk you are taking with your precious jars, you would like to know: how many jars does this box contain?

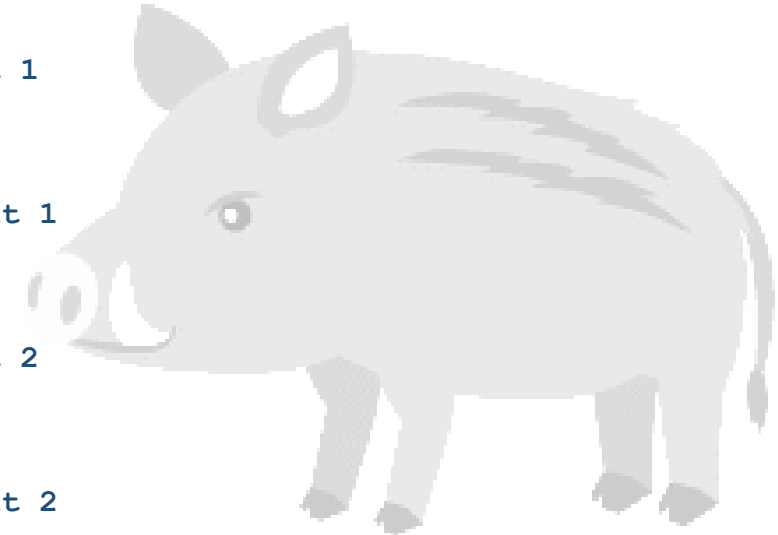
Input Format

The input consists of:

- A line with two integers n ($1 \le n \le 10^{18}$), the number of jars that need to be packed, and k ($1 \le k \le 10^{18}$), the number of jars a single box can hold.

Output Format

Output the number of jars that the least filled box contains.



Sample Input 1

```
10 3
```

Sample Output 1

```
2
```

Sample Input 2

```
16 4
```

Sample Output 2

```
4
```

Sample Input 3

```
1 2
```

Sample Output 3

```
1
```