Codeforces Sponsored by Telegram

# Codeforces Round #407 (Div. 2)

## A. Anastasia and pebbles

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

Anastasia loves going for a walk in Central Uzhlyandian Park. But she became uninterested in simple walking, so she began to collect Uzhlyandian pebbles. At first, she decided to collect all the pebbles she could find in the park.

She has only **two pockets**. She can put at most $k$ pebbles in each pocket at the same time. There are $n$ different pebble types in the park, and there are $w_i$ pebbles of the $i$-th type. Anastasia is very responsible, so she never mixes pebbles of different types in same pocket. However, she can put different kinds of pebbles in different pockets at the same time. Unfortunately, she can't spend all her time collecting pebbles, so she can collect pebbles from the park only once a day.

Help her to find the minimum number of days needed to collect all the pebbles of Uzhlyandian Central Park, taking into consideration that Anastasia can't place pebbles of different types in same pocket.

### Input

The first line contains two integers $n$ and $k$ $(1 \le n \le 10^5, 1 \le k \le 10^9)$ — the number of different pebble types and number of pebbles Anastasia can place in one pocket.

The second line contains $n$ integers $w_1, w_2, ..., w_n$ $(1 \le w_i \le 10^4)$ — number of pebbles of each type.

### Output

The only line of output contains one integer — the minimum number of days Anastasia needs to collect all the pebbles.

### Examples

input

```
3 2
2 3 4
```

output

```
3
```

input

```
5 4
3 1 8 9 7
```

output

```
5
```

### Note

In the first sample case, Anastasia can collect all pebbles of the first type on the first day, of second type — on the second day, and of third type — on the third day.

Optimal sequence of actions in the second sample case:

- In the first day Anastasia collects $8$ pebbles of the third type.
- In the second day she collects $8$ pebbles of the fourth type.
- In the third day she collects $3$ pebbles of the first type and $1$ pebble of the fourth type.
- In the fourth day she collects $7$ pebbles of the fifth type.
- In the fifth day she collects $1$ pebble of the second type.

# B. Masha and geometric depression

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

Masha really loves algebra. On the last lesson, her strict teacher Dvastan gave she new exercise.

You are given geometric progression $b$ defined by two integers $b_1$ and $q$. Remind that a geometric progression is a sequence of integers $b_1, b_2, b_3, \ldots$, where for each $i > 1$ the respective term satisfies the condition $b_i = b_{i-1} \cdot q$, where $q$ is called the common ratio of the progression. Progressions in Uzhlyandia are unusual: both $b_1$ and $q$ **can equal $0$**. Also, Dvastan gave Masha $m$ "bad" integers $a_1, a_2, \ldots, a_m$, and an integer $l$.

Masha writes all progression terms one by one onto the board (including repetitive) while condition $|b_i| \le l$ is satisfied ($|x|$ means absolute value of $x$). There is an exception: if a term equals one of the "bad" integers, Masha skips it (doesn't write onto the board) and moves forward to the next term.

But the lesson is going to end soon, so Masha has to calculate how many integers will be written on the board. In order not to get into depression, Masha asked you for help: help her calculate how many numbers she will write, or print "`inf`" in case she needs to write infinitely many integers.

### Input

The first line of input contains four integers $b_1, q, l, m$ ($-10^9 \le b_1, q \le 10^9$, $1 \le l \le 10^9$, $1 \le m \le 10^5$) — the initial term and the common ratio of progression, absolute value of maximal number that can be written on the board and the number of "bad" integers, respectively.

The second line contains $m$ distinct integers $a_1, a_2, \ldots, a_m$ ($-10^9 \le a_i \le 10^9$) — numbers that will never be written on the board.

### Output

Print the only integer, meaning the number of progression terms that will be written on the board if it is finite, or "`inf`" (without quotes) otherwise.

#### Examples

input

```
3 2 30 4
6 14 25 48
```

output

```
3
```

input

```
123 1 2143435 4
123 11 -5453 141245
```

output

```
0
```

input

```
123 1 2143435 4
54343 -13 6 124
```

output

```
inf
```

### Note

In the first sample case, Masha will write integers $3, 12, 24$. Progression term $6$ will be skipped because it is a "bad" integer. Terms bigger than $24$ won't be written because they exceed $l$ by absolute value.

In the second case, Masha won't write any number because all terms are equal $123$ and this is a "bad" integer.

In the third case, Masha will write infinitely integers $123$.

# C. Functions again

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

Something happened in Uzhlyandia again... There are riots on the streets... Famous Uzhlyandian superheroes Shean the Sheep and Stas the Giraffe were called in order to save the situation. Upon the arriving, they found that citizens are worried about maximum values of the Main Uzhlyandian Function $f$, which is defined as follows:

In the above formula, $1 \le l < r \le n$ must hold, where $n$ is the size of the Main Uzhlyandian Array $a$, and $|x|$ means absolute value of $x$. But the heroes skipped their math lessons in school, so they asked you for help. Help them calculate the maximum value of $f$ among all possible values of $l$ and $r$ for the given array $a$.

### Input

The first line contains single integer $n$ ($2 \le n \le 10^5$) — the size of the array $a$.

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

### Output

Print the only integer — the maximum value of $f$.

#### Examples

input

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

output

```
3
```

input

```
4
1 5 4 7
```

output

```
6
```

### Note

In the first sample case, the optimal value of $f$ is reached on intervals $[1, 2]$ and $[2, 5]$.

In the second case maximal value of $f$ is reachable only on the whole array.

# D. Weird journey

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little boy Igor wants to become a traveller. At first, he decided to visit all the cities of his motherland — Uzhlyandia.

It is widely known that Uzhlyandia has $n$ cities connected with $m$ bidirectional roads. Also, there are no two roads in the country that connect the same pair of cities, but roads starting and ending in the same city can exist. Igor wants to plan his journey beforehand. Boy thinks a path is *good* if the path goes over $m$ - $2$ roads twice, and over the other $2$ exactly once. The good path can start and finish in any city of Uzhlyandia.

Now he wants to know how many different good paths are in Uzhlyandia. Two paths are considered different if the sets of roads the paths goes over exactly once differ. Help Igor — calculate the number of good paths.

### Input

The first line contains two integers $n, m$ $(1 \le n, m \le 10^6)$ — the number of cities and roads in Uzhlyandia, respectively.

Each of the next $m$ lines contains two integers $u$ and $v$ $(1 \le u, v \le n)$ that mean that there is road between cities $u$ and $v$.

It is guaranteed that no road will be given in the input twice. That also means that for every city there is no more than one road that connects the city to itself.

### Output

Print out the only integer — the number of good paths in Uzhlyandia.

#### Examples

input

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

output

```
6
```

input

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

output

```
0
```

input

```
2 2
1 1
1 2
```

output

```
1
```

### Note

In first sample test case the good paths are:

- $2 \to 1 \to 3 \to 1 \to 4 \to 1 \to 5$ ,
- $2 \to 1 \to 3 \to 1 \to 5 \to 1 \to 4$ ,
- $2 \to 1 \to 4 \to 1 \to 5 \to 1 \to 3$ ,
- $3 \to 1 \to 2 \to 1 \to 4 \to 1 \to 5$ ,
- $3 \to 1 \to 2 \to 1 \to 5 \to 1 \to 4$ ,
- $4 \to 1 \to 2 \to 1 \to 3 \to 1 \to 5$ .

There are good paths that are same with displayed above, because the sets of roads they pass over once are same:

- $2 \to 1 \to 4 \to 1 \to 3 \to 1 \to 5$ ,
- $2 \to 1 \to 5 \to 1 \to 3 \to 1 \to 4$ ,
- $2 \to 1 \to 5 \to 1 \to 4 \to 1 \to 3$ ,
- $3 \to 1 \to 4 \to 1 \to 2 \to 1 \to 5$ ,
- $3 \to 1 \to 5 \to 1 \to 2 \to 1 \to 4$ ,
- $4 \to 1 \to 3 \to 1 \to 2 \to 1 \to 5$ ,
- and all the paths in the other direction.

Thus, the answer is $6$.

In the second test case, Igor simply can not walk by all the roads.

In the third case, Igor walks once over every road.

# E. The Great Mixing

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

Sasha and Kolya decided to get drunk with Coke, again. This time they have $k$ types of Coke. $i$-th type is characterised by its carbon dioxide concentration . Today, on the party in honour of Sergiy of Vancouver they decided to prepare a glass of Coke with carbon dioxide concentration . The drink should also be tasty, so the glass can contain only integer number of liters of each Coke type (some types can be not presented in the glass). Also, they want to minimize the total volume of Coke in the glass.

Carbon dioxide concentration is defined as the volume of carbone dioxide in the Coke divided by the total volume of Coke. When you mix two Cokes, the volume of carbon dioxide sums up, and the total volume of Coke sums up as well.

Help them, find the minimal natural number of liters needed to create a glass with carbon dioxide concentration . Assume that the friends have unlimited amount of each Coke type.

### Input

The first line contains two integers $n$, $k$ $(0 \le n \le 1000,\ 1 \le k \le 10^6)$ — carbon dioxide concentration the friends want and the number of Coke types.

The second line contains $k$ integers $a_1, a_2, \ldots, a_k$ $(0 \le a_i \le 1000)$ — carbon dioxide concentration of each type of Coke. Some Coke types can have same concentration.

### Output

Print the minimal natural number of liter needed to prepare a glass with carbon dioxide concentration , or `-1` if it is impossible.

#### Examples

input

```
400 4
100 300 450 500
```

output

```
2
```

input

```
50 2
100 25
```

output

```
3
```

### Note

In the first sample case, we can achieve concentration using one liter of Coke of types and : .

In the second case, we can achieve concentration using two liters of type and one liter of type: .