Codeforces Round #476 (Div. 2) [Thanks, Telegram!]

A. Paper Airplanes

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

To make a paper airplane, one has to use a rectangular piece of paper. From a sheet of standard size you can make $$$s$$$ airplanes.

A group of $$$k$$$ people decided to make $$$n$$$ airplanes each. They are going to buy several packs of paper, each of them containing $$$p$$$ sheets, and then distribute the sheets between the people. Each person should have enough sheets to make $$$n$$$ airplanes. How many packs should they buy?

Input
The only line contains four integers $$$k$$$, $$$n$$$, $$$s$$$, $$$p$$$ ($$$1 \le k, n, s, p \le 10^4$$$) — the number of people, the number of airplanes each should make, the number of airplanes that can be made using one sheet and the number of sheets in one pack, respectively.

Output
Print a single integer — the minimum number of packs they should buy.

Examples

input
5 3 2 3
output
4

input
5 3 100 1
output
5

Note
In the first sample they have to buy $$$4$$$ packs of paper: there will be $$$12$$$ sheets in total, and giving $$$2$$$ sheets to each person is enough to suit everyone's needs.

In the second sample they have to buy a pack for each person as they can't share sheets.

B. Battleship

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Arkady is playing Battleship. The rules of this game aren't really important.

There is a field of $$$n \times n$$$ cells. There should be exactly one $$$k$$$-decker on the field, i. e. a ship that is $$$k$$$ cells long oriented either horizontally or vertically. However, Arkady doesn't know where it is located. For each cell Arkady knows if it is definitely empty or can contain a part of the ship.

Consider all possible locations of the ship. Find such a cell that belongs to the maximum possible number of different locations of the ship.

Input
The first line contains two integers $$$n$$$ and $$$k$$$ ($$$1 \le k \le n \le 100$$$) — the size of the field and the size of the ship.

The next $$$n$$$ lines contain the field. Each line contains $$$n$$$ characters, each of which is either '#' (denotes a definitely empty cell) or '.' (denotes a cell that can belong to the ship).

Output
Output two integers — the row and the column of a cell that belongs to the maximum possible number of different locations of the ship.

If there are multiple answers, output any of them. In particular, if no ship can be placed on the field, you can output any cell.

Examples

input

```
4 3
#..#
#.#.
....
.###
```

output

```
3 2
```

input

```
10 4
#....##...
.#...#....
..#..#..#.
...#.#....
.#..##.#..
.....#...#
...#.##...
.#...#.#..
.....#..#.
...#.#...#
```

output

```
6 1
```

input

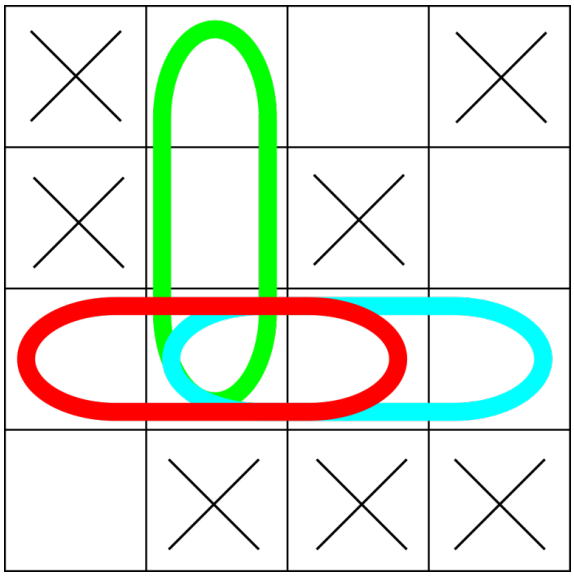
```
19 6
##.............###
#......#####.....##
.....#########.....
....###########....
...#############...
..###############..
.#################.
.#################.
.#################.
.#################.
#####....##....####
####............###
####............###
#####...####...####
.#####..####..#####
...###........###..
....###########....
.........##........
#.................#
```

output

```
1 8
```

Note

The picture below shows the three possible locations of the ship that contain the cell $$$(3, 2)$$$ in the first sample.



C. Greedy Arkady

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

$$$k$$$ people want to split $$$n$$$ candies between them. Each candy should be given to exactly one of them or be thrown away.

The people are numbered from $$$1$$$ to $$$k$$$, and Arkady is the first of them. To split the candies, Arkady will choose an integer $$$x$$$ and then give the first $$$x$$$ candies to himself, the next $$$x$$$ candies to the second person, the next $$$x$$$ candies to the third person and so on in a cycle. The leftover (the remainder that is not divisible by $$$x$$$) will be thrown away.

Arkady can't choose $$$x$$$ greater than $$$M$$$ as it is considered greedy. Also, he can't choose such a small $$$x$$$ that some person will receive candies more than $$$D$$$ times, as it is considered a slow splitting.

Please find what is the maximum number of candies Arkady can receive by choosing some valid $$$x$$$.

Input
The only line contains four integers $$$n$$$, $$$k$$$, $$$M$$$ and $$$D$$$ ($$$2 \le n \le 10^{18}$$$, $$$2 \le k \le n$$$, $$$1 \le M \le n$$$, $$$1 \le D \le \min{(n, 1000)}$$$, $$$M \cdot D \cdot k \ge n$$$) — the number of candies, the number of people, the maximum number of candies given to a person at once, the maximum number of times a person can receive candies.

Output
Print a single integer — the maximum possible number of candies Arkady can give to himself.

Note that it is always possible to choose some valid $$$x$$$.

Examples

input
20 4 5 2
output
8

input
30 9 4 1
output
4

Note
In the first example Arkady should choose $$$x = 4$$$. He will give $$$4$$$ candies to himself, $$$4$$$ candies to the second person, $$$4$$$ candies to the third person, then $$$4$$$ candies to the fourth person and then again $$$4$$$ candies to himself. No person is given candies more than $$$2$$$ times, and Arkady receives $$$8$$$ candies in total.

Note that if Arkady chooses $$$x = 5$$$, he will receive only $$$5$$$ candies, and if he chooses $$$x = 3$$$, he will receive only $$$3 + 3 = 6$$$ candies as well as the second person, the third and the fourth persons will receive $$$3$$$ candies, and $$$2$$$ candies will be thrown away. He can't choose $$$x = 1$$$ nor $$$x = 2$$$ because in these cases he will receive candies more than $$$2$$$ times.

In the second example Arkady has to choose $$$x = 4$$$, because any smaller value leads to him receiving candies more than $$$1$$$ time.

D. Single-use Stones

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

A lot of frogs want to cross a river. A river is $$$w$$$ units width, but frogs can only jump $$$l$$$ units long, where $$$l < w$$$. Frogs can also jump on lengths shorter than $$$l$$$. but can't jump longer. Hopefully, there are some stones in the river to help them.

The stones are located at integer distances from the banks. There are $$$a_i$$$ stones at the distance of $$$i$$$ units from the bank the frogs are currently at. Each stone can only be used once by one frog, after that it drowns in the water.

What is the maximum number of frogs that can cross the river, given that then can only jump on the stones?

Input
The first line contains two integers $$$w$$$ and $$$l$$$ ($$$1 \le l < w \le 10^5$$$) — the width of the river and the maximum length of a frog's jump.

The second line contains $$$w - 1$$$ integers $$$a_1, a_2, \ldots, a_{w-1}$$$ ($$$0 \le a_i \le 10^4$$$), where $$$a_i$$$ is the number of stones at the distance $$$i$$$ from the bank the frogs are currently at.

Output
Print a single integer — the maximum number of frogs that can cross the river.

Examples

input
10 5 0 0 1 0 2 0 0 1 0
output

```
3
```

input

```
10 3
1 1 1 1 2 1 1 1 1
```

output

```
3
```

Note

In the first sample two frogs can use the different stones at the distance $$$5$$$, and one frog can use the stones at the distances $$$3$$$ and then $$$8$$$.

In the second sample although there are two stones at the distance $$$5$$$, that does not help. The three paths are: $$$0 \to 3 \to 6 \to 9 \to 10$$$, $$$0 \to 2 \to 5 \to 8 \to 10$$$, $$$0 \to 1 \to 4 \to 7 \to 10$$$.

E. Short Code

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

Arkady's code contains $$$n$$$ variables. Each variable has a unique name consisting of lowercase English letters only. One day Arkady decided to shorten his code.

He wants to replace each variable name with its non-empty prefix so that these new names are still unique (however, a new name of some variable can coincide with some old name of another or same variable). Among such possibilities he wants to find the way with the smallest possible total length of the new names.

A string $$$a$$$ is a prefix of a string $$$b$$$ if you can delete some (possibly none) characters from the end of $$$b$$$ and obtain $$$a$$$.

Please find this minimum possible total length of new names.

Input

The first line contains a single integer $$$n$$$ ($$$1 \le n \le 10^5$$$) — the number of variables.

The next $$$n$$$ lines contain variable names, one per line. Each name is non-empty and contains only lowercase English letters. The total length of these strings is not greater than $$$10^5$$$. The variable names are distinct.

Output

Print a single integer — the minimum possible total length of new variable names.

Examples

input

```
3
codeforces
codehorses
code
```

output

```
6
```

input

```
5
abba
abb
ab
aa
aacada
```

output

```
11
```

input

```
3
telegram
digital
resistance
```

output

```
3
```

Note

In the first example one of the best options is to shorten the names in the given order as "`cod`", "`co`", "`c`".

In the second example we can shorten the last name to "`aac`" and the first name to "`a`" without changing the other names.