Codeforces Round #355 (Div. 2)

A. Vanya and Fence

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

Vanya and his friends are walking along the fence of height h and they do not want the guard to notice them. In order to achieve this the height of each of the friends should not exceed h. If the height of some person is greater than h he can bend down and then he surely won't be noticed by the guard. The height of the i-th person is equal to a_i.

Consider the width of the person walking as usual to be equal to 1, while the width of the bent person is equal to 2. Friends want to talk to each other while walking, so they would like to walk in a single row. What is the minimum width of the road, such that friends can walk in a row and remain unattended by the guard?

Input

The first line of the input contains two integers n and h ($1 \le n \le 1000$, $1 \le h \le 1000$) — the number of friends and the height of the fence, respectively.

The second line contains n integers a_i ($1 \le a_i \le 2h$), the i-th of them is equal to the height of the i-th person.

Output

Print a single integer — the minimum possible valid width of the road.

Examples

input

```
3 7
4 5 14
```

output

```
4
```

input

```
6 1
1 1 1 1 1 1
```

output

```
6
```

input

```
6 5
7 6 8 9 10 5
```

output

```
11
```

Note

In the first sample, only person number 3 must bend down, so the required width is equal to $1+1+2=4$.

In the second sample, all friends are short enough and no one has to bend, so the width $1+1+1+1+1+1=6$ is enough.

In the third sample, all the persons have to bend, except the last one. The required minimum width of the road is equal to $2+2+2+2+2+1=11$.

B. Vanya and Food Processor

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

Vanya smashes potato in a vertical food processor. At each moment of time the height of the potato in the processor doesn't exceed h and the processor smashes k centimeters of potato each second. If there are less than k centimeters remaining, than during this second processor smashes all the remaining potato.

Vanya has n pieces of potato, the height of the i-th piece is equal to a_i. He puts them in the food processor one by one starting from the piece number 1 and finishing with piece number n. Formally, each second the following happens:

1. If there is at least one piece of potato remaining, Vanya puts them in the processor one by one, until there is not enough space for the next piece.
2. Processor smashes k centimeters of potato (or just everything that is inside).

Provided the information about the parameter of the food processor and the size of each potato in a row, compute how long will it take for all the potato to become smashed.

Input

The first line of the input contains integers n, h and k ($1 \le n \le 100\,000$, $1 \le k \le h \le 10^9$) — the number of pieces of potato, the height of the food processor and the amount of potato being smashed each second, respectively.

The second line contains n integers a_i ($1 \le a_i \le h$) — the heights of the pieces.

Output

Print a single integer — the number of seconds required to smash all the potatoes following the process described in the problem statement.

Examples

input

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

output

```
5
```

input

```
5 6 3
5 5 5 5 5
```

output

```
10
```

input

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

output

```
2
```

Note

Consider the first sample.

1. First Vanya puts the piece of potato of height 5 into processor. At the end of the second there is only amount of height 2 remaining inside.
2. Now Vanya puts the piece of potato of height 4. At the end of the second there is amount of height 3 remaining.
3. Vanya puts the piece of height 3 inside and again there are only 3 centimeters remaining at the end of this second.
4. Vanya finally puts the pieces of height 2 and 1 inside. At the end of the second the height of potato in the processor is equal to 3.
5. During this second processor finally smashes all the remaining potato and the process finishes.

In the second sample, Vanya puts the piece of height 5 inside and waits for 2 seconds while it is completely smashed. Then he repeats the same process for 4 other pieces. The total time is equal to $2 \cdot 5 = 10$ seconds.

In the third sample, Vanya simply puts all the potato inside the processor and waits 2 seconds.

C. Vanya and Label

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

While walking down the street Vanya saw a label "Hide&Seek". Because he is a programmer, he used $\&$ as a bitwise AND for these two words represented as a integers in base 64 and got new word. Now Vanya thinks of some string s and wants to know the number of pairs of words of length $|s|$ (length of s), such that their bitwise AND is equal to s. As this number can be large, output it modulo $10^9 + 7$.

To represent the string as a number in numeral system with base 64 Vanya uses the following rules:

- digits from '`0`' to '`9`' correspond to integers from 0 to 9;
- letters from '`A`' to '`Z`' correspond to integers from 10 to 35;
- letters from '`a`' to '`z`' correspond to integers from 36 to 61;
- letter '`-`' correspond to integer 62;
- letter '`_`' correspond to integer 63.

Input

The only line of the input contains a single word s ($1 \leq |s| \leq 100\ 000$), consisting of digits, lowercase and uppercase English letters, characters '`-`' and '`_`'.

Output

Print a single integer — the number of possible pairs of words, such that their bitwise AND is equal to string s modulo $10^9 + 7$.

Examples

input
```
z
```
output
```
3
```

input
```
V_V
```
output
```
9
```

input
```
Codeforces
```
output
```
130653412
```

Note

For a detailed definition of bitwise AND we recommend to take a look in the corresponding article in Wikipedia.

In the first sample, there are 3 possible solutions:

1. $z\&\_ = 61\&63 = 61 = z$
2. $\_\&z = 63\&61 = 61 = z$
3. $z\&z = 61\&61 = 61 = z$

D. Vanya and Treasure

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

Vanya is in the palace that can be represented as a grid $n \times m$. Each room contains a single chest, an the room located in the i-th row and j-th columns contains the chest of type a_{ij}. Each chest of type $x \le p - 1$ contains a key that can open any chest of type $x + 1$, and all chests of type 1 are not locked. There is exactly one chest of type p and it contains a treasure.

Vanya starts in cell $(1, 1)$ (top left corner). What is the minimum total distance Vanya has to walk in order to get the treasure? Consider the distance between cell (r_1, c_1) (the cell in the row r_1 and column c_1) and (r_2, c_2) is equal to $|r_1 - r_2| + |c_1 - c_2|$.

Input

The first line of the input contains three integers n, m and p ($1 \le n, m \le 300$, $1 \le p \le n \cdot m$) — the number of rows and columns in the table representing the palace and the number of different types of the chests, respectively.

Each of the following n lines contains m integers a_{ij} ($1 \le a_{ij} \le p$) — the types of the chests in corresponding rooms. It's guaranteed that for each x from 1 to p there is at least one chest of this type (that is, there exists a pair of r and c, such that $a_{rc} = x$). Also, it's guaranteed that there is exactly one chest of type p.

Output

Print one integer — the minimum possible total distance Vanya has to walk in order to get the treasure from the chest of type p.

Examples

input

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

output

```
5
```

input

```
3 3 9
1 3 5
8 9 7
4 6 2
```

output

```
22
```

input

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

output

```
11
```

E. Vanya and Balloons

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Vanya plays a game of balloons on the field of size $n \times n$, where each cell contains a balloon with one of the values 0, 1, 2 or 3. The goal is to destroy a cross, such that the product of all values of balloons in the cross is maximum possible. There are two types of crosses: normal and rotated. For example:

```
**o**
**o**
ooooo
**o**
**o**
```

or

```
o***o
*o*o*
**o**
*o*o*
o***o
```

Formally, the cross is given by three integers r, c and d, such that $d \le r, c \le n - d + 1$. The normal cross consists of balloons located in cells (x, y) (where x stay for the number of the row and y for the number of the column), such that $|x - r| \cdot |y - c| = 0$ and $|x - r| + |y - c| < d$. Rotated cross consists of balloons located in cells (x, y), such that $|x - r| = |y - c|$ and $|x - r| < d$.

Vanya wants to know the maximum possible product of the values of balls forming one cross. As this value can be large, output it modulo $10^9 + 7$.

Input

The first line of the input contains a single integer n ($1 \le n \le 1000$) — the number of rows and columns in the table with balloons.

The each of the following n lines contains n characters '0', '1', '2' or '3' — the description of the values in balloons.

Output

Print the maximum possible product modulo $10^9 + 7$. Note, that you are not asked to maximize the remainder modulo $10^9 + 7$, but to find the maximum value and print it this modulo.

Examples

input

```
4
1233
0213
2020
0303
```

output

```
108
```

input

```
5
00300
00300
33333
00300
00300
```

output

```
19683
```

input

```
5
00003
02030
00300
03020
30000
```

output

```
108
```

input

```
5
21312
10003
10002
10003
23231
```

output

```
3
```

input

```
5
12131
12111
12112
21311
21212
```

output

```
24
```

Note

In the first sample, the maximum product is achieved for a rotated cross with a center in the cell $(3, 3)$ and radius 1: $2 \cdot 2 \cdot 3 \cdot 3 \cdot 3 = 108$.