A. Reconnaissance

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

According to the regulations of Berland's army, a reconnaissance unit should consist of exactly two soldiers. Since these two soldiers shouldn't differ much, their heights can differ by at most d centimeters. Captain Bob has n soldiers in his detachment. Their heights are $a_1, a_2, ..., a_n$ centimeters. Some soldiers are of the same height. Bob wants to know, how many ways exist to form a reconnaissance unit of two soldiers from his detachment.

Ways $(1, 2)$ and $(2, 1)$ should be regarded as different.

Input

The first line contains two integers n and d $(1 \le n \le 1000, 1 \le d \le 10^9)$ — amount of soldiers in Bob's detachment and the maximum allowed height difference respectively. The second line contains n space-separated integers — heights of all the soldiers in Bob's detachment. These numbers don't exceed 10^9.

Output

Output one number — amount of ways to form a reconnaissance unit of two soldiers, whose height difference doesn't exceed d.

Sample test(s)

input

```
5 10
10 20 50 60 65
```

output

```
6
```

input

```
5 1
55 30 29 31 55
```

output

```
6
```

B. Borze

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

Ternary numeric notation is quite popular in Berland. To telegraph the ternary number the Borze alphabet is used. Digit 0 is transmitted as «`.`», 1 as «`-.`» and 2 as «`--`». You are to decode the Borze code, i.e. to find out the ternary number given its representation in Borze alphabet.

Input

The first line contains a number in Borze code. The length of the string is between 1 and 200 characters. It's guaranteed that the given string is a valid Borze code of some ternary number (this number can have leading zeroes).

Output

Output the decoded ternary number. It can have leading zeroes.

Sample test(s)

input

```
.-.--
```

output

```
012
```

input

```
--.
```

output

```
20
```

input

```
-..-.--
```

output

```
1012
```

C. Flea

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

It is known that fleas in Berland can jump only vertically and horizontally, and the length of the jump is always equal to s centimeters. A flea has found herself at the center of some cell of the checked board of the size $n \times m$ centimeters (each cell is 1×1 centimeters). She can jump as she wishes for an arbitrary number of times, she can even visit a cell more than once. The only restriction is that she cannot jump out of the board.

The flea can count the amount of cells that she can reach from the starting position (x, y). Let's denote this amount by $d_{x, y}$. Your task is to find the number of such starting positions (x, y), which have the maximum possible value of $d_{x, y}$.

Input

The first line contains three integers n, m, s $(1 \le n, m, s \le 10^6)$ — length of the board, width of the board and length of the flea's jump.

Output

Output the only integer — the number of the required starting positions of the flea.

Sample test(s)

input

```
2 3 1000000
```

output

```
6
```

input

```
3 3 2
```

output

```
4
```

D. Constellation

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

A star map in Berland is a checked field $n \times m$ squares. In each square there is or there is not a star. The favourite constellation of all Berland's astronomers is the constellation of the Cross. This constellation can be formed by any 5 stars so, that for some integer x (*radius of the constellation*) the following is true:

- the 2nd is on the same vertical line as the 1st, but x squares up
- the 3rd is on the same vertical line as the 1st, but x squares down
- the 4th is on the same horizontal line as the 1st, but x squares left
- the 5th is on the same horizontal line as the 1st, but x squares right

Such constellations can be very numerous, that's why they are numbered with integers from 1 on the following principle: when two constellations are compared, the one with a smaller radius gets a smaller index; if their radii are equal — the one, whose central star if higher than the central star of the other one; if their central stars are at the same level — the one, whose central star is to the left of the central star of the other one.

Your task is to find the constellation with index k by the given Berland's star map.

Input

The first line contains three integers n, m and k ($1 \le n, m \le 300$, $1 \le k \le 3 \cdot 10^7$) — height and width of the map and index of the required constellation respectively. The upper-left corner has coordinates $(1, 1)$, and the lower-right — (n, m). Then there follow n lines, m characters each — description of the map. j-th character in i-th line is «*», if there is a star in the corresponding square, and «.» if this square is empty.

Output

If the number of the constellations is less than k, output `-1`. Otherwise output 5 lines, two integers each — coordinates of the required constellation. Output the stars in the following order: central, upper, lower, left, right.

Sample test(s)

input

```
5 6 1
....*.
...***
....*.
..*...
.***..
```

output

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

input

```
5 6 2
....*.
...***
....*.
..*...
.***..
```

output

```
-1
```

input

```
7 7 2
...*...
.......
...*...
*.***.*
...*...
.......
...*...
```

output

```
4 4
1 4
7 4
4 1
4 7
```

E. Hide-and-Seek

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

Victor and Peter are playing hide-and-seek. Peter has hidden, and Victor is to find him. In the room where they are playing, there is only one non-transparent wall and one double-sided mirror. Victor and Peter are points with coordinates (x_v, y_v) and (x_p, y_p) respectively. The wall is a segment joining points with coordinates $(x_{w,1}, y_{w,1})$ and $(x_{w,2}, y_{w,2})$, the mirror — a segment joining points $(x_{m,1}, y_{m,1})$ and $(x_{m,2}, y_{m,2})$.

If an obstacle has a common point with a line of vision, it's considered, that the boys can't see each other with this line of vision. If the mirror has a common point with the line of vision, it's considered, that the boys can see each other in the mirror, i.e. reflection takes place. The reflection process is governed by laws of physics — the angle of incidence is equal to the angle of reflection. The incident ray is in the same half-plane as the reflected ray, relative to the mirror. I.e. to see each other Victor and Peter should be to the same side of the line, containing the mirror (see example 1). If the line of vision is parallel to the mirror, reflection doesn't take place, and the mirror isn't regarded as an obstacle (see example 4).

Victor got interested if he can see Peter, while standing at the same spot. Help him solve this problem.

Input

The first line contains two numbers x_v and y_v — coordinates of Victor.

The second line contains two numbers x_p and y_p — coordinates of Peter.

The third line contains 4 numbers $x_{w,1}, y_{w,1}, x_{w,2}, y_{w,2}$ — coordinates of the wall.

The forth line contains 4 numbers $x_{m,1}, y_{m,1}, x_{m,2}, y_{m,2}$ — coordinates of the mirror.

All the coordinates are integer numbers, and don't exceed 10^4 in absolute value. It's guaranteed, that the segments don't have common points, Victor and Peter are not on any of the segments, coordinates of Victor and Peter aren't the same, the segments don't degenerate into points.

Output

Output `YES`, if Victor can see Peter without leaving the initial spot. Otherwise output `NO`.

Sample test(s)

input
```
-1 3
1 3
0 2 0 4
0 0 0 1
```
output
```
NO
```

input
```
0 0
1 1
0 1 1 0
-100 -100 -101 -101
```
output
```
NO
```

input
```
0 0
1 1
0 1 1 0
-1 1 1 3
```
output
```
YES
```

input
```
0 0
10 0
100 100 101 101
1 0 3 0
```
output
```
YES
```

Codeforces (c) Copyright 2010-2015 Mike Mirzayanov
The only programming contests Web 2.0 platform