

## 7395 Connect

The “Manhattan” or “taxi” distance between any two points on a grid is defined as the shortest distance measured along the grid lines. For example, given a one-unit grid and the points A, B and C on the diagram shown as Figure 1, the distances are as follows:

A to B: 3  
 A to C: 4  
 B to C: 1  
 B to A: 3  
 C to A: 4  
 C to B: 1

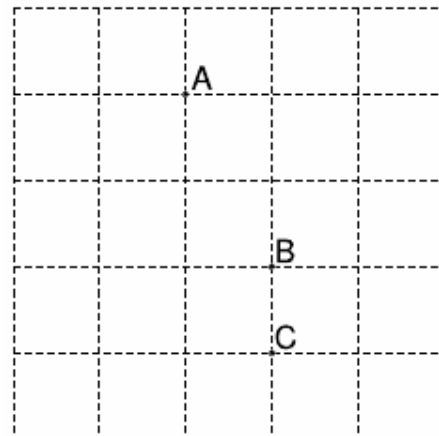


Figure 1. Example “Manhattan Distance” Grid.

An interesting question then is: Given  $n$  distinct points on a grid, how many distinct shortest distances are there between every pair of points? In the example above, the possible distances are (1, 3, 4) so the answer to the question is “three distinct distances.” (As more points are added, eventually that number will not increase as fast as  $n$ .) Your team is to write a program that will determine the number of distinct distances between selected points on a grid.

### Input

The input file contains several test cases, each of them as described below.

Input to your program will be a map represented by an array of characters terminated by end-of-file. The points are ‘x’ characters and spaces are ‘0’ characters. The map is a square: the number of characters on each line will be equal to the number of lines. The maximum size of the map is  $80 \times 80$ . The distance between each adjacent character counts as one unit of distance. The grid in Figure 1 would be represented as:

```
000000
00x000
000000
000x00
000x00
000000
```

### Output

For each test case, your program’s output is to be the count of distinct shortest distances between all pairs of points on the map. Print the count on a line with no unnecessary leading zeroes and no leading or trailing whitespace.

### Sample Input

```
000000000000000000000000
000x00x00000000000000000
0000000000000000000000x0
```

```
0x0000000000000000x00
000000000000000000x
00000000000000000x00
00000000000000000000
00000000000x00000000
00000000000000000000
0x0000000000000000x000
0000x0000000000000000
00x0000000000x0000000
00x000000x000x0x0000
0x00000000x000x00000
00000000000x00000000
0000000000000000000x
00000000000000000000
00000000000000000000
0000x0000x0000000000
00000000000x00000000
```

### Sample Output

30