

WC '99 P2 - Mission Impossible

Time limit: 1.0s **Memory limit:** 16M

Woburn Challenge 1999

Due to the unprecedented popularity of the last Mission Impossible movie, and especially the scene with Tom Cruise falling from the vent, John Woo has done what Hollywood does best - repeat the scene with minor variations in the sequel. So Ethan Hunt finds himself again having to crawl through the vents of a high-security building and like most movie-sized vents, it fits a full-person along with all his gear. This time however, he has to drop into a room full of explosives in order to retrieve the top-secret solutions to Woburn Challenge 2000. So naturally, he wants to know the exact position of all devices in the room before he drops in. In preparation, he takes the following 2 items: a thermal imager and 1 bottle of hair gel (hey, he has to look good for his entrance, right?) The thermal imager is to scan the room from the ventilation ducts (i.e. from above) to determine how many heat signatures there are (it seems that all the explosives give off heat).

Input Specification

Your mission (and you must accept it) is to write the program that will determine, from a thermal scan, how many explosives are in the room. Here are some assumptions you can make to simplify your job. Assume that no 2 objects are connected thermally. Also assume that all the explosives give off heat throughout their bulk (but objects need not be solid) and that only these objects give off heat. Here are some specs on your imager. It will provide you with a pixel image of the room. Each pixel will be represented by a number from 0 to 9, with (0 = no heat, 5 = heat given off by the brainwork in this contest, 9 = heat from the infernal fires of hell).

Note that each image will be at most of size 20×20 . At the start of every input will be a line containing the number of columns / rows (respectively). There may be many images in the file; `-1` denotes the end of input.

Output Specification

For each test case, output the number of explosives are in the room.

Sample Input

```
5 5
01200
00002
00002
00000
00000
00000
-1 -1
```

Sample Output
