

MENDO

System for competitions in informatics 2026

English

Македонски

Home

Learn C++

Competitions

Help

» 739782 tested submissions | 22759 users «

Home

Training

Piano

Chat OFFLINE

AI

Login

Username

Password

☐ Remember Me

Submit

Password forgotten?

Navigation

Register

Learn C++

Algorithms

Training

Competitions

Forums

Wiki

Help

Report Bug

Piano

The organizers of JBOI 2015 have spent a lot of time planning the closing ceremony, and they want to make sure everybody enjoys the evening. That is why they've invited a world famous Macedonian piano player Simon Trpchevski to play at the ceremony. There is just one problem – the ceremony is in a huge one floor building, and the available pianos are far from the stage.

In the entire building, there are exactly N pianos ($N \leq 3$). The organizers of JBOI know the location of every single piano, and the location where the piano needs to be taken. A piano has a rectangular shape and it can only be pushed in one of four available directions (left, right, up and down). Due to the large weight of a piano, organizers can't rotate it in any way.

Simon is asking for your help. All of the information that the organizers have is displayed on a rectangular map of the building. Can you write a program that will calculate which one of the pianos can be taken to the stage in the smallest number of steps, and without moving the other pianos in the process?

Input

The first line of input contains two integers W and H ($1 \leq W, H \leq 50$), the width and height of the rectangular map of the building.

Each of the next H lines will contain W characters. The possible characters are '#' (an obstacle, where no piece of a piano can ever move), '.' (a free location), '1'-'3' (describing the location of the pianos), and 'F' (the final location of the piano, on the stage). The location of the first piano will be defined with 1's, the location of the second piano (if there is one) with 2's, and the location of the third piano (if there is one) with 3's.

A piano has a rectangular shape of size at most 5 x 5 (at most 5 cells wide and at most 5 cells high). All pianos on the map will have the same shape and orientation. The final location will have the same shape and orientation as the pianos.

Output

The output consists of two integers in two separate lines: which piano is closest to the venue (1, 2, or 3), and the distance from that piano to the venue (the number of required steps). As described above, the possible steps are pushing a piano to the left, right, up or down. If there are multiple pianos which are closest to the venue (i.e. they have the same shortest distance), output any of them.

It will always be possible to bring at least one of the pianos to the venue. It's possible that some pianos cannot be taken to the venue of the closing ceremony, i.e. they can be blocked by '#' cells or by other pianos on the map.

Constraints

Time limit: 1 second
Memory limit: 256 megabytes

Examples

input

output

13 7

1

111.....

14

111.....

..#####

.222#.....

.222#.....

.....FFF.

....#....FFF.

input

output

5 5

1

..1..

10

..#.#2

..#.#

.....

..##F

https://mendo.mk/Task.do?id=598

1/2

Scoring

In test cases worth at least 30% of all points, the size of a piano will be 1x1 (pianos will take exactly one cell on the map). The second test case presented above fits this constraint.

Submit Test

Submit your code

Solution language:

Automatic Detection...

▼

Upload solution (option 1):

Select

Paste your code (option 2):

CodeMirror

Submit!

Copyright © 2026 Mendo Judge Team | All Rights Reserved
Supported by [CSM](#) | Powered by [Struts](#) | [XHTML 1.0](#) | [CSS 2.0](#) | [FAMFAMFAM](#)
» This website (project) is dedicated to the memory of [Nikola Postolov](#) «

https://mendo.mk/Task.do?id=598

2/2