Exercises — Lakes of Finland

version **#7be580532266ed398481e31366afcc24b1950c2a**

The way is lit. The path is clear.
We require only the strength to follow it.

Assistants C/Unix 2023 <assistants@tickets.assistants.epita.fr>

Contents

[*]https://intra.assistants.epita.fr

File Tree

```
lakes/
├─ lake.in
├─ lakes-example.c
├─ lakes.c   (to submit)
└─ lakes.h   (to submit)
```

Authorized functions : You are only allowed to use the following functions

- malloc(3)
- free(3)

Authorized headers : You are only allowed to use the functions defined in the following headers

- err.h
- errno.h
- assert.h
- stddef.h

Compilation : Your code must compile with the following flags

- -std=c99 -pedantic -Werror -Wall -Wextra -Wvla

1 Goal

A geographical map is represented as a 2D array of characters. Each character can be either '#' (ground) or '.' (water). The purpose of this exercise is to count the number of lakes on the map.

A lake is a water area made of connected water delimited by ground. Two water cells are said to be connected if they are vertically or horizontally adjacent. That means that two water cell diagonally adjacent are not connected:

```
##
.. connected

#.
#. connected

#.
.# not connected
```

In order to make the problem simpler, there will not be any map with water on the border. The minimum number of lakes is 0, which means there can be a map without any lake.

You have to implement the following function:

```c
int lakes(char **map, int width, int height);
```

- `map` represents the map
- `width` and `height` are the map dimensions

You will not have to handle error cases.

2 Example

The given `lakes-example.c` results to:

```
42sh$ cat -e lake.in
30 20$
##############################$
#######################...####$
###..#####.....#######....####$
##....#####.....#######..#####$
##...#####.....###############$
###..#######...###############$
###################...########$
##################....########$
######...##########....#######$
#####.....################..##$
####.....#################..##$
####....#####..###....#####.##$
##########.....##......#######$
#######.......####.....#######$
####...####..#####.....#######$
###.....###########....#######$
###..........#################$
####..........################$
#####################.########$
##############################$
42sh$ gcc -Wall -Wextra -Werror -std=c99 -pedantic lakes.c lakes-example.c -o lakes
42sh$ ./lakes lake.in | cat -e
10$
```

The way is lit. The path is clear. We require only the strength to follow it.