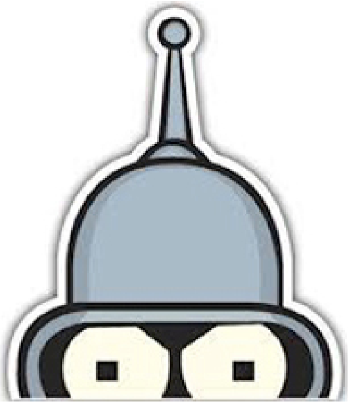
B1 - Elementary Programming in C

B-CPE-110

BSQ

find the Biggest SQuare



BSQ

binary name: bsq
language: C
compilation: via Makefile, including re, clean and fclean rules

- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Authorized Functions

The only system calls allowed are the following ones:

- open
- read
- write
- close
- malloc
- free
- stat

THE PROJECT

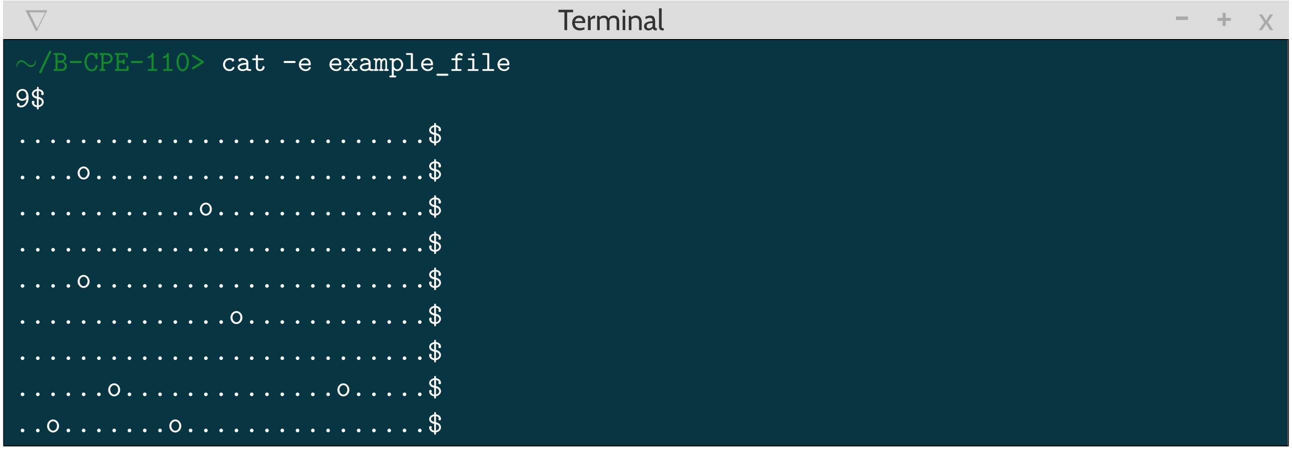
You must find the largest possible square on a board while avoiding obstacles.
The board can be generated two ways. The first one is by reading a file passed as the program's argument.
The file is valid if it is respecting those constraints:

- its first line contains the number of lines on the board (and only that),
- "." (representing an empty place) and "o" (representing an obstacle) are the only two characters for the other lines,
- all of the lines are of the same length (except the first one),
- it contains at least one line,
- each line is terminated by '\n'.

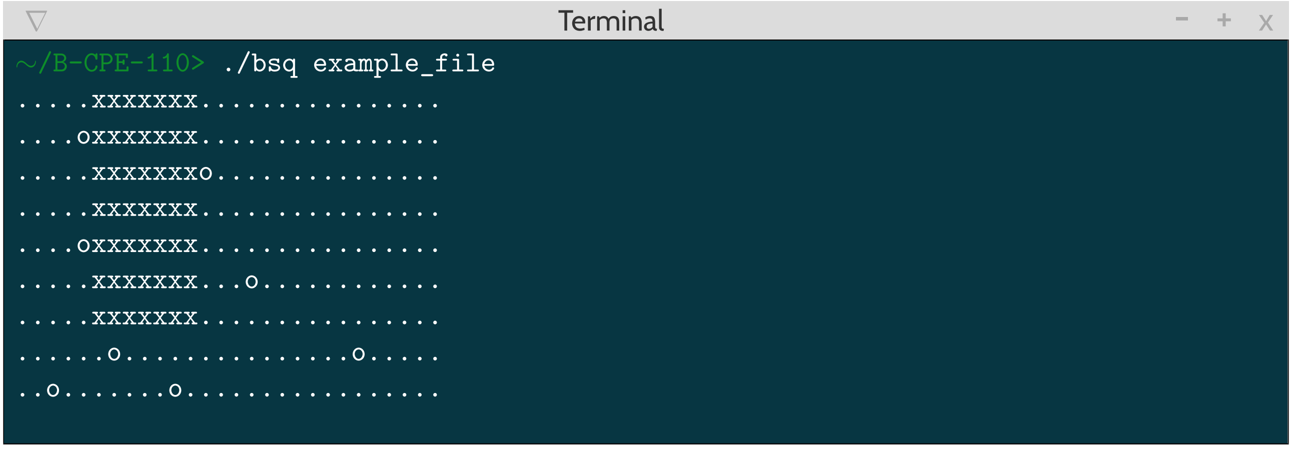
You program must print the board, with some "." replaced by "x" to represent the largest square you found.

If ever there are several solutions, you have to represent only the highest square. If they are still several solutions, choose the square to the left.

EXAMPLES



```
~/B-CPE-110> cat -e example_file
9$
...........................$
....o......................$
............o..............$
...........................$
....o......................$
..............o............$
...........................$
......o..............o.....$
..o.......o................$
```



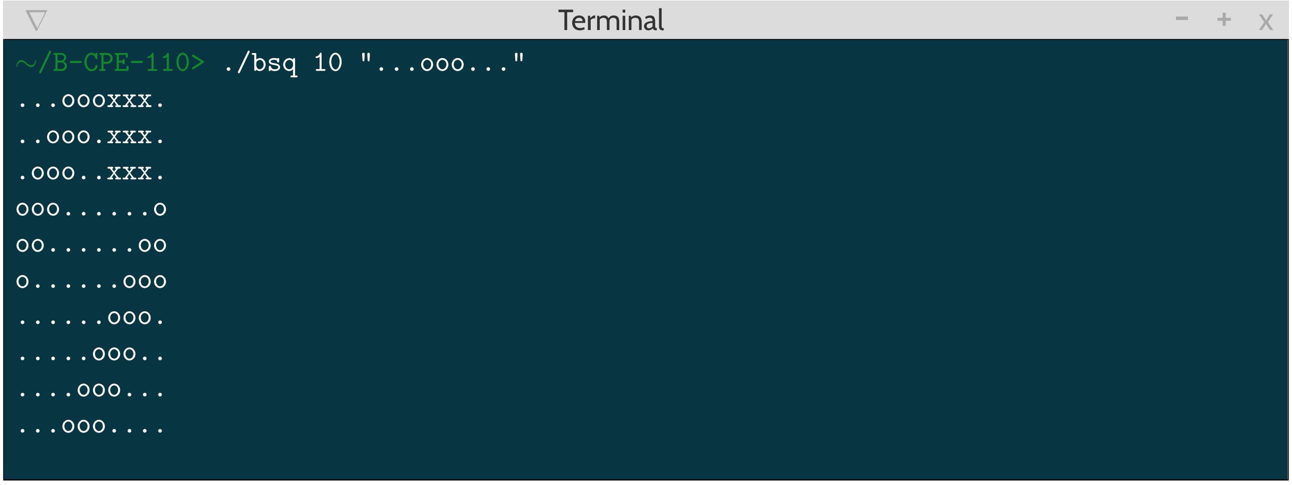
```
~/B-CPE-110> ./bsq example_file
.....xxxxxxx................
....oxxxxxxx................
.....xxxxxxxo...............
.....xxxxxxx................
....oxxxxxxx................
.....xxxxxxx...o............
.....xxxxxxx................
......o...............o.....
..o.......o.................
```

It’s definitely a square, even if visually it doesn’t look like one.

You only need to **read** from the given files. Meaning that your program should work even if you don’t have the permissions to write on the files.

Generating your own

The second way to obtain a board is to generate one based on given parameters. The parameters will be a number, representing the width and height of the board, and a pattern that will be repeated line by line along the board.
You will print the solved board.



```
~/B-CPE-110> ./bsq 10 "...ooo..."
...oooxxx.
..ooo.xxx.
.ooo..xxx.
ooo......o
oo......oo
o......ooo
......ooo.
.....ooo..
....ooo...
...ooo....
```