B1- Elementary Programming in C

B-CPE-110

BSQ

find the Biggest SQuare

BSQ

find the Biggest SQuare

binary name:	bsq
repository name:	CPE_BSQ_$ACADEMICYEAR
repository rights:	ramassage-tek
language:	C
group size:	1
compilation:	via Makefile, including re, clean and fclean rules

> (!)
> - Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
> - 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).

You must find the largest possible square on a board while avoiding obstacles.
The board is represented by a file passed as the program's argument, 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 allowed characters for the other lines,

- all of the lines will be the same length (except the first one),

- there will always be 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.

Authorized Functions

The only system calls allowed are the following ones:
- open
- read
- write
- close
- malloc
- free
- exit
- stat

Examples

> (!) It's definitely a square, even if visually it doesn't look like one

Perl Board Generator

Here is a Perl script that will enable you to create boards. You're welcome.

```perl
#!/usr/bin/perl -w

if ((scalar @ARGV) != 3)
{
        print "program x y density\n";
        exit;
}

my $x = $ARGV[0];
my $y = $ARGV[1];
my $density = $ARGV[2];
my $i = 0;
my $j = 0;

print $y . "\n";

while ($i < $y)
{
        $j = 0;
        while ($j < $x)
        {
                if (int(rand($y)*2) < $density)
                {
                        print "o";
                }
                else
                {
                        print ".";
                }
                $j++;
        }
        print "\n";
        $i++;
}
```