Elementary programming

BSQ

Astek in charge wiart_m@epitech.eu

Abstract: This document is the subject of the BSQ Elementary programming project.

Contents

Chapter I

Instructions

- Your code must comply with the norm
- Your sources must be turned in on the `CPE_year_bsq` repository (ex : CPE_2014_bsq).

⚠ Pay attention to the permissions of your files and directories ...

Chapter II

Subject

- The goal of the project is to find the biggest possible square on a board, while avoiding obstacles.
- The board will be given to you in a file, passed as argument to your program.
- The board is composed of lines of '.' and 'o'.
- The first line of the board is a number indicating the number of lines in the board.
- All lines have the same length.
- The board will always be a rectangle.
- There will always be at least a line of at least one cell.
- At the end of every line, there is a '\n'.
- Example :

```
foo@bar>cat example_file
9
...........................
....o......................
............o..............
...........................
....o......................
...............o...........
...........................
......o..............o.....
..o.......o................
foo@bar>
```

- The goal of the program is to remplace the '.' by 'x' to represent the biggest square possible.

- When several solutions are possible, we will choose to represent the top-leftmost square.

- Example :

```
foo@bar>./bsq example\_file
.....xxxxxxx...............
....oxxxxxxx...............
.....xxxxxxxo..............
.....xxxxxxx...............
....oxxxxxxx...............
.....xxxxxxx...o...........
.....xxxxxxx...............
......o..............o.....
..o.......o................
foo@bar>
```

Even if it does not visually look like a square, it is a square ...

Chapter III

Perl board generator

- The following perl script allows you to create boards:

```
#!/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++;
}
```

Chapter IV

Allowed functions

- open
- read
- write
- close
- exit
- malloc
- free