C Piscine's final Project

BSQ

42 staff staff@42.fr

Summary: Will you find the biggest square?

Contents

Chapter 1

Foreword

Extracts from Life, The Universe, and Everything:

« Important facts from Galactic history, number one:
[Reproduced from the Siderial Daily Mentioner's Book of popular Galactic History.]
The night sky over the planet Krikkit is the least interesting sight in the entire Universe. »

« The Krikkit Wars belonged to the ancient past of the Galaxy, and Zaphod had spent most of his early history lessons plotting how he was going to have sex with the girl in the cybercubicle next to him, and since his teaching computer had been an integral part of this plot it had eventually had all its history circuits wiped and replaced with an entirely different set of ideas which had then resulted in it being scrapped and sent to a home for Degenerate Cybermats, whither it was followed by the girl who had inadvertently fallen deeply in love with the unfortunate machine, with the result [a] that Zaphod never got near her and [b] that he missed out on a period of ancient history that would have been of inestimable value to him at this moment. »

« The game you know as cricket, [Slartibartfast] said, and his voice still seemed to be wandering lost in subterranean passages, is just one of those curious freaks of racial memory which can keep images alive in the mind aeons after their true significance has been lost in the mists of time. Of all the races on the Galaxy, only the English could possibly revive the memory of the most horrific wars ever to sunder the Universe and transform it into what I'm afraid is generally regarded as an incomprehensibly dull and pointless game. »

« Although it has been said that on Earth alone in our Galaxy is Krikkit [or cricket] treated as fit subject for a game, and that for this reason the Earth has been shunned, this does only apply to our Galaxy, and more specifically to our dimension. In some of the higher dimensions they feel they can more or less please themselves, and have been playing a peculiar game called Brockian Ultra-Cricket for whatever their transdimensional equivalent of billions of years is. »

Did you know that other versions of Krikkit exist? The Brockian Simple-Qricket, for instance, is played on a field of variable size. The only constant is its shape: a perfect square. Throughout the ages, many techniques have been tried to use the available space as efficiently as possible. The BSQ, one of the first eco-friendly games, requires total respect of the environment. No obstacle can be moved to create the playground, be it a tree, a stone, a road sign, a starship anchor, or, during a memorable match between the Legally Administrated Vogons' team and the Titans' Titans team, an entire galactic sector. The irony lying, of course, in the fate of the Galactic Sector ZZ9 Plural Z Alpha shortly after the Vogons' victory.

Chapter 2

Subject

- The biggest square:
    - The aim of this project is to find the biggest square on a map, avoiding obstacles.
    - A file containing the map will be provided. It'll have to be passed as an argument for your program.
    - The first line of the map contains information on how to read the map:
        * The number of lines on the map;
        * The "empty" character;
        * The "obstacle" character;
        * The "full" character.
    - The map is made up of '**"empty" characters**', lines and '**"obstacle" characters**'.
    - The aim of the program is to replace '**"empty" characters**' by '**"full" characters**' in order to represent the biggest square possible.
    - If more than one solution exists, we'll choose to represent the square that's closest to the top of the map, then the one that's most to the left.

- Definition of a valid map:
  - All lines must have the same length.
  - There's at least one line of at least one character.
  - At each end of line, there's a line break.
  - The characters on the map can only be those introduced in the first line.
  - In case of an invalid map, your program should display **map error** on the error output followed by a line break. Your program will then move on to the next map.
- Here's an example of how it should work :

```
%>cat example_file
9.ox
.............................
....o........................
............o................
.............................
....o........................
...............o............
.............................
......o..............o......
..o.......o..................
%>./bsq example_file
.....xxxxxxx................
....oxxxxxxx................
.....xxxxxxxo...............
.....xxxxxxx................
....oxxxxxxx................
.....xxxxxxx...o............
.....xxxxxxx................
......o..............o......
..o.......o..................
%>
```

It is a square indeed. Even though it might not look like it visually.

Chapter 3

Instructions

- The executable must be called **bsq** and be present in the main directory.
- You must respect the Norm.
- You may only use methods learnt during the Piscine.
- The submission directory must have an author file containing your logins :

```
$>cat author
login_1:login_2
$>
```

- Your program must handle 1 to n files as parameters.
- If your program receives more than one map as argument, all the solutions and errors that are printed should be separated by a line break.
- Should there be no passed arguments, your program must be able to read on the standard input.
- You should have a valid Makefile that'll compile your project.
- You may only use the following functions : **exit**, **open**, **close**, **write**, **read**, **malloc** and **free**.

Chapter 4

Grading

- BSQ evaluation info :
    - We'll start by testing functionality (10 points total). Your program must be functional.
    - If you get all of those 10 points, we'll then check optimisation. Two things will be evaluated here: execution speed (5 points) and memory usage (5 points).
        * Those two aspects will be checked **only** if the first part has been validated.
        * All BSQs that have passed the first part of this evaluation will be competing for the two optimisation tests.
        * The best BSQ will be rewarded with all optimisation points.
        * Other BSQs will get extra points depending on how they're ranked: the faster the execution, and/or the the least used memory, the more points it gets; the slower the execution and/or the more used memory, the fewer points it gets.
        * TL;DR: Only one group should get all 20 points. You'll be graded depending on your BSQ's rank compared to other BSQs.

  Good luck to all of you!

Chapter 5

Annex

- Perl map generator

```
#!/usr/bin/perl

use warnings;
use strict;

die "program x y density" unless (scalar(@ARGV) == 3);

my ($x, $y, $density) = @ARGV;

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