



Projet Piscine C

BSQ

42 staff staff@42.fr

Résumé: Trouveras-tu le plus grand carré ?

Table des matières

I	Préambule	2
II	Sujet	4
III	Consignes	6
IV	Notation	7
V	Annexe	8

Chapitre I

Préambule

Extraits de 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. »

Saviez-vous que d'autres variantes du Krikkit existent ? Le Brockian Simple-Qricket a la particularité de se jouer sur un terrain de superficie variable. La seule constante est sa forme : un carré parfait. À travers les âges, de nombreuses techniques ont été adoptées afin de pouvoir optimiser l'espace disponible. Le BSQ, l'un des premiers jeux écoresponsable, demande en effet de respecter l'environnement. Aucun obstacle ne peut être déplacé pour

la création du terrain de jeu, que ce soit arbre, pierre, panneau de signalisation, ancre de vaisseau ou, lors d'une rencontre particulière entre l'équipe des Administrés Légaux Vogons et celles des Titans de Titans, secteur galactique entier. L'ironie étant le destin du Galactic Sector ZZ9 Plural Z Alpha peu après la victoire des Vogons.

Chapitre II

Sujet

- Le plus grand carré :
 - Il s'agit de trouver le plus grand carré possible sur un plateau en évitant des obstacles.
 - Un plateau vous est transmis dans un fichier passé en argument du programme.
 - La première ligne du plateau contient les informations pour lire la carte :
 - Le nombre de lignes du plateau ;
 - Le caractère "vide" ;
 - Le caractère "obstacle" ;
 - Le caractère "plein".
 - Le plateau est composé de lignes de 'caractères "vide"' et de 'caractères "obstacle"'.
 - Le but du programme est de remplacer les 'caractères "vide"' par des 'caractères "plein"' pour représenter le plus grand carré possible.
 - Dans le cas où plusieurs solutions existent, on choisira de représenter le carré le plus en haut à gauche.

- Définition d'une carte valide :
 - Toutes les lignes doivent avoir la même longueur.
 - Il y a au moins une ligne d'au moins une case.
 - À la fin de chaque ligne, il y a un retour à la ligne.
 - Les caractères présent dans la carte doivent être uniquement ceux présenté à la première ligne.
 - En cas de carte invalide, vous afficherez sur la sortie d'erreur : `map error` suivi d'un retour à la ligne. Le programme passera ensuite au traitement du plateau suivant.
- Exemple de fonctionnement :

```
%>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.....
%>
```



C'est bien un carré. Même si cela n'y ressemble pas visuellement.

Chapitre III

Consignes

- L'exécutable doit s'appeler **bsq** et se trouver dans le répertoire principal.
- Votre projet doit être à la Norme.
- Vous ne pouvez utiliser que les éléments vus durant votre Piscine.
- Le répertoire doit avoir un fichier auteur dans lequel vous devez mettre vos logins :

```
$>cat auteur  
login_1:login_2  
$>
```

- Le programme peut prendre de 1 à n fichiers en paramètre.
- Si aucun argument n'est passé, le programme lira sur l'entrée standard (il n'y aura qu'une seule carte dans ce cas).
- Votre Makefile devra compiler le projet.
- Vous ne pouvez utiliser que les fonctions **exit**, **open**, **close**, **write**, **read**, **malloc** et **free**.
- Vous pouvez poser vos questions sur le forum.

Chapitre IV

Notation

- La notation de la BSQ s'effectue en deux temps :
 - Nous commençons par tester **las funcionalidades** (sur 10 points). Votre programme doit fonctionner.
 - La deuxième partie est **la optimización** du code. Elle est notée sur deux parties indépendantes : la vitesse d'exécution (5 points) la mémoire utilisé (5 points).
 - Elle n'est vérifiée que si toute la première partie est correcte.
 - Tous les BSQ ayant validé l'intégralité de la première partie seront en compétition sur les deux tests d'optimisations.
 - Le meilleur projet remportera l'intégralité des points de cette partie.
 - Les projets suivants remporteront des points en fonction de leur classement : pour la vitesse d'exécution, les plus rapides auront plus de points que les plus lents. Pour la mémoire, ceux qui ont utilisé le moins de mémoire auront plus de points que ceux qui en utilisent plus.

Bon courage à tous !

Chapitre V

Annexe

- Générateur de plateaux perl

```
#!/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";
}
```