



# Projet Fin l Piscine C

BSQ

42 staff [st\\_ff@42.fr](mailto:st_ff@42.fr)

*Résumé: Trouver s-tu le plus gr nd c rré ?*

*Version: 8*

# Table des matières

I	Consignes	2
II	Préambule	4
III	Sujet	6
IV	Annexe	10
V	Rendu et peer-evaluation	11

# Chapitre I

## Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Relisez bien le sujet avant de rendre vos exercices. tout moment le sujet peut changer.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices.
- Vos exercices seront corrigés par vos camarades de piscine.
- En plus de vos camarades, vous serez corrigés par un programme appelé la Moulinette.
- La Moulinette est très stricte dans sa notation. Elle est totalement automatisée. Il est impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les surprises.
- La Moulinette n'est pas très ouverte d'esprit. Elle ne cherche pas à comprendre le code qui ne respecte pas la Norme. La Moulinette utilise le programme **norminette** pour vérifier la norme de vos fichiers. Comprendre par là qu'il est stupide de rendre un code qui ne passe pas la **norminette**.
- L'utilisation d'une fonction interdite est un cas de triche. Toute triche est sanctionnée par la note de -42.
- Vous ne devrez rendre une fonction `main()` que si nous vous demandons un programme.
- La Moulinette compile avec les flags `-Wall -Wextra -Werror`, et utilise `cc`.
- Si votre programme ne compile pas, vous aurez 0.
- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Votre manuel de référence s'appelle **Google / m n / Internet / ....**
- Lisez attentivement les exemples. Ils pourraient bien requérir des choses qui ne

sont pas autrement précisées dans le sujet...

- Réfléchissez. Par pitié, par Odin ! Nom d'une pipe.

# Chapitre II

## 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. »

« lthough 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 III

## Sujet

Nom du programme	bsq
Fichiers de rendu	Makefile et tous les fichiers nécessaires
Makefile	Yes
Arguments	Fichier(s) dans le(s)quel(s) lire le carré
Fonctions externes autorisées	open, close, read, write, malloc, free, exit
Libft autorisée	Ne concerne pas ce projet
Description	Écrivez un programme qui dessine et affiche à l'écran le plus grand carré possible dans la zone donnée

- 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 puis le plus à gauche.

Votre programme doit accepter de 1 à n paramètres.

Lorsque votre programme reçoit plusieurs paramètres en argument, chaque solution ou `m p error` doit être suivi d'un line break.

Si il n'y a aucun argument, votre fonction doit lire le carré sur l'entrée standard.

Vous devez rendre un Makefile qui compile votre projet. Votre Makefile ne doit pas relink.



- 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.

La carte est invalide si un caractère manque sur la première ligne, ou si deux caractères (parmis vide, plein, et obstacles) sont identiques.

Les caractères peuvent être n'importe quel caractère imprimable, même des nombres.

En cas de carte invalide, vous afficherez sur la sortie d'erreur : `m p 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.....
....XXXXXXXX.....
.....XXXXXXXO.....
....XXXXXXX.....
....XXXXXXXX.....
....XXXXXXXX.....
.....XXXXXXX...O.....
.....XXXXXXX.....
.....O.....O.....
..O.....O.....
%>
```



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

# Chapitre IV

## nnexe

- Générateur de plateaux perl

```
#!/usr/bin/perl

use warnings;
use strict;

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

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

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";
}
```

# Chapitre V

## Rendu et peer-evaluation

Rendez votre travail sur votre dépôt `Git` comme d'habitude. Seul le travail présent sur votre dépôt sera évalué en soutenance. Vérifiez bien les noms de vos dossiers et de vos fichiers afin que ces derniers soient conformes aux demandes du sujet.

Vu que votre travail ne sera pas évalué par un programme, organisez vos fichiers comme bon vous semble du moment que vous rendez les fichiers obligatoires et respectez les consignes du sujet.



Vous ne devez rendre uniquement les fichiers demandés par le sujet de ce projet.