



ASTEK



Programmation élémentaire

BSQ

Responsable Astek astek_resp@epitech.eu

Abstract:



Table des matières

.1	Consignes	2
.2	Sujet	3
.3	Générateur de plateaux perl	5
.4	Fonctions autorisées	6



.1 Consignes

- Votre code doit être à la norme.
- Les sources doivent être rendues sur le dépôt SVN :
svn+ssh ://kscm@koala-rendus.epitech.net/bsq-2016-login_x



.2 Sujet

- Il s'agit de trouver le plus grand carré possible sur un plateau en évitant des obstacles.
- Le plateau vous est transmis dans un fichier passé en argument du programme.
- Le plateau est composé de lignes de '.' et de 'o'.
- La première ligne du plateau est un nombre qui indique le nombre de lignes du plateau.
- Toutes les lignes ont la même longueur.
- Le plateau sera toujours un rectangle.
- Il y a au moins une ligne d'au moins une case.
- À la fin de chaque ligne il y a un '\n'.
- Exemple :

```
1  foo@bar>cat example_file
2  9
3  .....
4  ....O.....
5  .....O.....
6  .....
7  ....O.....
8  .....O.....
9  .....
10 .....O.....O.....
11 ..O.....O.....
12 foo@bar>
```

- Le but du programme est de remplacer les '.' par des 'x' pour représenter le plus grand carré possible.
- Dans le cas où il y en a plusieurs solutions, on choisira de représenter le carré le plus en haut à gauche.
- Exemple :

```
1  foo@bar>./bsq example\_file
2  ....XXXXXXXX.....
3  ....OXXXXXXXX.....
4  ....XXXXXXXXO.....
5  ....XXXXXXXX.....
6  ....OXXXXXXXX.....
7  ....XXXXXXXX...O.....
```



```
8      .....xxxxxxx.....  
9      .....O.....O.....  
10     ..O.....O.....  
11     foo@bar>
```



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



.3 Générateur de plateaux perl

- Le script perl suivant permet de créer des plateaux

```
1 #!/usr/bin/perl -w
2
3 if ((scalar @ARGV) != 3)
4 {
5     print "program x y density\n";
6     exit;
7 }
8
9 my $x = $ARGV[0];
10 my $y = $ARGV[1];
11 my $density = $ARGV[2];
12 my $i = 0;
13 my $j = 0;
14
15 print $y . "\n";
16
17 while ($i < $y)
18 {
19     $j = 0;
20     while ($j < $x)
21     {
22         if (int(rand($y)*2) < $density)
23         {
24             print "o";
25         }
26         else
27         {
28             print ".";
29         }
30         $j++;
31     }
32     print "\n";
33     $i++;
34 }
```



.4 Fonctions autorisées

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