



Turbo 4.2

BSQ

Sumário: Quarto mini-projeto do Turbo 4.2.

Conteúdo

I	Preâmbulo	2
II	Exercício	4
III	Instruções	6
IV	Avaliação	7
V	Anexo	8

Capítulo I

Preâmbulo

Extraído de A Vida, O Universo e Tudo Mais:

« Fatos importantes extraídos da História Galáctica, número um: (Reproduzido do Livro de História Galáctica Popular do Siderial Daily Mentioner's.)

O céu noturno do planeta Krikkit é a vista menos interessante de todo o Universo. »

«As Guerras de Krikkit pertenciam ao passado antigo da Galáxia, e Zaphod havia gasto a maioria de suas aulas de história antiga elaborando um plano para transar com a garota que ocupava o cibercubículo ao lado.

Uma vez que o computador responsável por suas aulas era parte integral desse plano, ele eventualmente teve todos os seus circuitos de história apagados e substituídos por um conjunto completamente diferente de idéias. Como resultado disso, o computador foi desmontado e enviado para um abrigo para Cibertrastes Degenerados. Ele foi seguido pela garota, que havia inadvertidamente se apaixonado pela pobre máquina, coisa que, por sua vez, resultou em (a) Zaphod nunca ter conseguido nada com ela e (b) ele ter deixado de estudar um período de história antiga que teria um valor inestimável para ele naquele momento. »

« O jogo que você conhece como críquete — disse [Slartibartfast], com uma voz que parecia ainda vagar por subterrâneos — é apenas uma dessas peculiaridades da memória racial, capaz de manter algumas imagens vivas na mente séculos após seu verdadeiro sentido ter se perdido nas névoas do tempo. De todas as raças da Galáxia, apenas os ingleses seriam capazes de reviver a memória da mais terrível das guerras que já cindiram o Universo e transformá-la naquilo que, lamento dizer, é visto como um jogo incompreensivelmente chato e sem sentido. »

« Ainda que tenham dito que, em toda a nossa Galáxia, apenas na Terra o Krikkit (ou críquete) é tratado como um assunto adequado para um jogo e que, por este motivo, ela tenha sido posta à parte, isso só se aplica à nossa Galáxia e, mais especificamente, à nossa dimensão. Em algumas das dimensões mais elevadas as pessoas acreditam que é possível se divertir pelo menos um pouco, e eles têm jogado algo muito peculiar, chamado Ultracríquete Broquiano, durante seja lá qual for o equivalente transdimensional deles para bilhões de anos. »

Você sabia que existem outras versões do Krikkit? O Brockian Simple-Qricket, por exemplo, é jogado em um campo de tamanho variável. A única constante é a sua forma: um quadrado perfeito. Ao longo dos tempos, muitas técnicas foram tentadas para usar o espaço disponível da forma mais eficiente possível. O BSQ, um dos primeiros jogos ecologicamente corretos, exige total respeito ao meio ambiente. Nenhum obstáculo pode ser movido para criar o playground, seja uma árvore, uma pedra, uma placa de trânsito, uma âncora de nave ou, durante uma partida memorável entre a equipe dos Vogons Legalmente Administrados e a equipe dos Titãs dos Titãs, um setor galáctico inteiro. A ironia está, é claro, no destino do Setor Galáctico ZZ9 Plural Z Alpha logo após a vitória dos Vogons em relação ao meio

ambiente. Nenhum obstáculo pode ser movido para criar o playground, seja uma árvore, uma pedra, uma placa de trânsito, uma âncora de nave ou, durante uma partida memorável entre a equipe dos Vogons Legalmente Administrados e a equipe dos Titãs dos Titãs, uma galáxia inteira setor. A ironia está, é claro, no destino do Setor Galáctico ZZ9 Plural Z Alpha logo após a vitória dos Vogons.

Capítulo II

Exercício

- O maior quadrado:
 - O objetivo deste projeto é encontrar o maior quadrado do mapa, evitando obstáculos.
 - Será fornecido um arquivo contendo o mapa. Este arquivo terá de ser passado como argumento para o seu programa.
 - A primeira linha do mapa contém informações sobre como ler o mapa:
 - O número de linhas no mapa;
 - O caractere "vazio";
 - O caractere "obstáculo";
 - O caractere "completo".
 - O mapa é composto por 'caracteres "vazios"', linhas e 'caracteres "obstáculo" '.
 - O objetivo do programa é substituir 'caracteres "vazios"' por 'caracteres "cheios"' para representar o maior quadrado possível.
 - Se existir mais de uma solução, escolheremos representar o quadrado que está mais próximo do topo do mapa, e depois aquele que está mais à esquerda.

- Definição de um mapa válido:
 - Todas as linhas devem ter o mesmo comprimento.
 - Há pelo menos uma linha de pelo menos uma quadrado.
 - Em cada final de linha, há uma quebra de linha.
 - Os caracteres do mapa só podem ser aqueles introduzidos na primeira linha.
 - No caso de um mapa inválido, seu programa deve exibir erro de mapa no erro saída seguida por uma quebra de linha. E então o seu programa deve passar para o próximo mapa.
- Aqui está um exemplo de como deve funcionar:

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



É um quadrado mesmo. Mesmo que não pareça visualmente.

Capítulo III

Instruções

- O executável deve se chamar `bsq` e estar presente no diretório principal.
- Você deve respeitar a Norma.
- Você só pode usar métodos aprendidos durante a Piscina/Basecamp.
- O diretório de submissão deve ter um arquivo de autor contendo seus logins:

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

- Seu programa deve tratar de 1 a n arquivos como parâmetros.
- Caso seu programa receba mais de um mapa como argumento, todas as soluções e erros que são impressos devem ser separados por uma quebra de linha.
- Caso não haja argumentos passados, seu programa deve ser capaz de ler na entrada (*input*) padrão.
- Você deve ter um Makefile válido que compilará seu projeto.
- Você só pode usar as seguintes funções: `exit`, `open`, `close`, `write`, `read`, `malloc` e `free`.
- Você pode fazer perguntas no discord

Capítulo IV

Avaliação

- Informações de avaliação BSQ:
 - Começaremos testando as funcionalidades (de 10 pontos). Seu programa deve ser funcional.
 - Se você obtiver todos esses 10 pontos, verificaremos a otimização do seu código. Duas coisas serão avaliadas aqui: velocidade de execução (5 pontos) e uso de memória (5 pontos).
 - Esses dois aspectos **só** serão verificados se a primeira parte tiver sido validada.

Boa sorte a todos vocês!

Capítulo V

Anexo

- Gerador de mapa em 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";
}
```