

Trabalho 02 - Labirinto Sádico

Data de entrega: 19/10/2022 19hs

Importante:

- **Não** olhe códigos de outros ou da internet. Exceto o que é fornecido. Também não mostre ou publique o seu. Não utilize o repl.it ou qualquer outra plataforma que exponha seu código.
- Em caso de plágio, fraude ou tentativa de burlar o sistema será aplicado nota 0 na disciplina aos envolvidos.
- Alguns alunos podem ser solicitados para explicar com detalhes a implementação.
- Passar em todos os testes do run.codes não é garantia de tirar a nota máxima. Sua nota ainda depende do cumprimento das especificações do trabalho, qualidade do código, clareza dos comentários, boas práticas de programação e entendimento da matéria demonstrada em possível reunião.
- Apenas o capitão da dupla deverá submeter o código na plataforma run.codes.

Este trabalho deverá ser feito em duplas (as mesmas) e implementado em linguagem C.

Considere o seguinte jogo: Um jogador é largado (vestido) em uma sala qualquer de um conjunto de várias salas, dispostas em um grid n por n , sendo que algumas dessas salas estão bloqueadas. Em cada sala o jogador pode ir para a sala ao norte, ao sul, ao leste ou ao oeste, desde que a sala exista e não esteja bloqueada. Apenas a sala mais ao norte e mais ao oeste (mais acima e à esquerda) tem uma saída para o mundo exterior (essa sala nunca estará bloqueada). O objetivo do jogador é chegar na saída. Os desenvolvedores do jogo adicionaram uma propriedade para ajudar o jogador a sair desse labirinto pelo caminho mais curto possível, em cada sala está tocando uma sirene, cujo volume é mais alto quanto mais salas você precisa andar até chegar na saída. Ou seja, se o jogador consegue andar só 1 sala para chegar na saída, a sirene toca com volume 1, mas se ele está em uma sala em que precisará andar pelo menos 20 salas até chegar na saída a sirene toca com volume 20. O fator sádico é que a sirene é, na verdade, um sertanejo universitário, tão irritante que se o jogador ficar preso no labirinto por muito tempo, ele certamente ficará maluco. Também é possível que existam salas a partir das quais não exista caminho para a saída, nesse caso o jogador está preso para sempre.

Nesse trabalho você deverá **adaptar o algoritmo de cálculo de distância entre cidades visto em aula**, para ajudar os desenvolvedores a descobrir qual o volume deverá tocar a sirene do tihoso em cada sala. Você **obrigatoriamente deverá usar a TAD Fila** que será fornecida. (Dica: você pode criar mais de uma fila)

Seu programa deverá ler da saída padrão do sistema um inteiro n indicando as dimensões do grid e depois n linhas, cada uma com n caracteres que formam a configuração do labirinto, o caractere '0' (zero) indica uma sala livre, e o caractere 'X' indica uma sala bloqueada. No final de cada linha não há espaços em branco e uma quebra de linha '\n' termina cada linha. Por exemplo, um grid 3 por 3 em que apenas a sala a leste (a direita) da sala com a saída está bloqueada, seria representada pela seguinte entrada:

```
3
0X0
000
000
```

O seu programa deverá imprimir n linhas cada uma com n inteiros, cada um seguido de um espaço em branco, e cada linha terminada com um '\n', com o volume que deve estar tocando em cada sala. Uma sala que está bloqueada ou da onde é impossível chegar na saída deverá tocar a sirene com volume n ao quadrado, indicando que a partir dali não há saída (em nenhum caminho válido o jogador precisará andar n ao quadrado salas, pense no motivo). Para o exemplo acima, a saída esperada é:

```
0 9 4
1 2 3
2 3 4
```

Você receberá alguns arquivos para auxiliar no seu desenvolvimento, incluindo a TAD Fila, você deverá **modificar e submeter apenas o arquivo main.c**. Não esqueça de liberar toda a memória alocada.

Para compilar múltiplos arquivos, o jeito que considero mais correto é o seguinte, Primeiramente compilar cada um dos arquivos .c:

```
gcc -Wall -pedantic-errors pilha.c -c
```

e

```
gcc -Wall -pedantic-errors main.c -c
```

Esses comandos vão gerar os arquivos pilha.o e main.o

O próximo passo é linkar esses arquivos (quando você estiver usando outras bibliotecas, também é nesse momento que você linka seu programa com elas).

```
gcc -Wall -pedantic-errors main.o pilha.o -o exec
```

Não deixe para a última semana!