

Projeto Final - A

Profa. Simone Aires

Grupo: 3 alunos no máximo

Data de entrega: 01/12/2022 23:59 – via Moodle

Instruções:

- (a) Os arquivos solicitados em cada questão devem ser colocados em um diretório cujo nome corresponde ao número da questão. Exemplos: 1/ e 2/.
- (b) Os diretórios com as soluções de cada questão (descritos acima) devem ser colocados em um arquivo compactado e nomeado com os RAs de cada membro da equipe (no máximo 3 alunos).

Exemplo:

1234-5678-9101112.tar.gz

1234-5678-9101112.rar

para uma equipe com os alunos de RA número 1234, 5678 e 9101112.

(c) Data da Entrega: 01/06/2022 23:59 – via Moodle

(d) **O professor reserva-se o direito de anular a questão/projeto se:** as especificações não forem rigorosamente atendidas, houver erros de compilação, extensão dos arquivos com formato diferente do pedido, elevado grau de similaridade entre códigos de equipes distintas, envio incorreto (i.e., fora do prazo ou por meio não estabelecido pelo professor), dentre outros.

(e) A Questão 2 – não precisa ser recursiva.

(f) **Apresentação Questão 2** – vídeo com **todos** da equipe.

- Orientações sobre o vídeo ao final do Projeto.

1) Apresente o código fonte de todos os algoritmos recursivos pedidos abaixo em uma biblioteca chamada recursividade.c. Apresente também um programa chamado main.c que exemplifica (como você desejar) o uso de cada uma dos procedimentos da biblioteca.

(a) Devolver a multiplicação através de soma de dois números inteiros; $a, b \in \mathbb{N}$

(b) Devolver a^b onde $a; b \in \mathbb{N}$.

(c) Devolver o fatorial de um número natural $n \in \mathbb{N}$

(d) Dado $n \in \mathbb{N}$, imprimir números pares que incidem no intervalo $[0; n]$

(e) Dada uma string, imprimir na tela o número de vezes que um caracter aparece na string dada. O caracter a ser procurado será informado pelo usuário.

(f) Devolver o índice i de um dado vetor v com n inteiros tal que $v[i] == k$ ou -1 caso tal índice inexista. k é dado.

(g) Devolver o produto da soma de todos os elementos de um dado vetor v com n valores reais. A partir da i -ésima chamada recursiva devem ser realizadas necessariamente duas outras chamadas recursivas. DICA: em cada chamada, divida o vetor em dois outros menores e combine a soma deles em uma só.

2) Rota de Fuga: Os irmãos João e Maria estão da Bruxa Keka, que corre atrás deles dentro da casa de doces. João e Maria acabam de entrar em uma sala retangular de largura N e comprimento M , dividida em $N \times M$ células (i, j) de área 1 ($1 \leq i \leq N$ e $1 \leq j \leq M$). Em algumas células (i, j) dessa sala, existem armários. A sala tem uma entrada na célula (X_e, Y_e) e uma saída na célula (X_s, Y_s) , que ficam em posições diferentes, nas bordas da sala.

A figura a seguir mostra uma possível configuração da sala, onde $N = 7$ e $M = 7$. A entrada fica na posição $(3, 7)$ (marcada com uma estrela) e a saída fica na posição $(5, 1)$ (marcada com um círculo). Os armários estão indicados em quadrados cinzas.

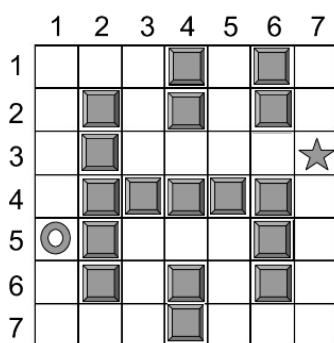


Figura 1

Sua tarefa é traçar a menor rota de fuga, para que os irmãos consigam escapar da Bruxa Keka.

Apresente o menor caminho a ser percorrido (posições i, j) da entrada até a saída, e o número total de células percorridas para completar o percurso.

A Figura 2 apresenta alguns percursos possíveis de saída. O menor caminho apresenta 11 células percorridas.

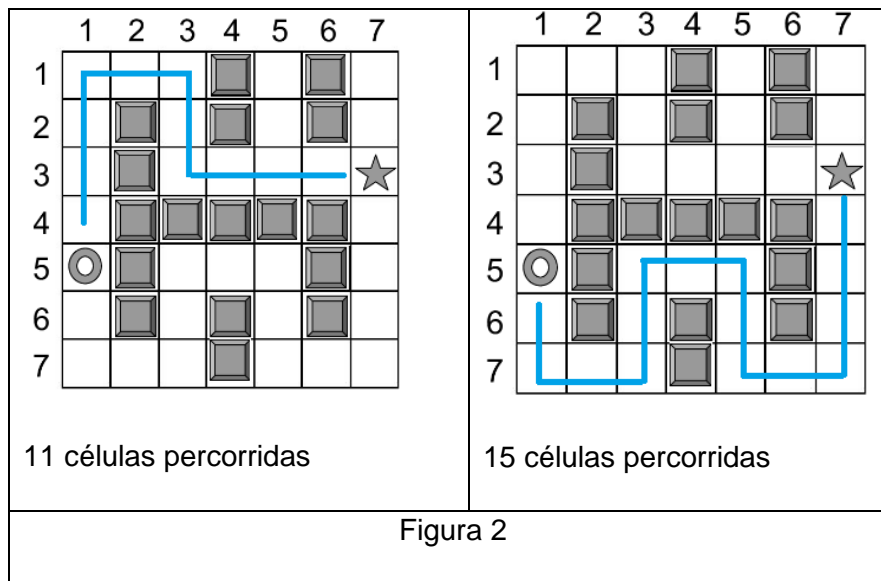


Figura 2

Entrada

A primeira linha contém dois inteiros N e M , referentes a largura e o comprimento da sala, respectivamente. A segunda linha contém dois inteiros X_e e Y_e , identificando a célula de entrada da sala (X_e, Y_e). A terceira linha contém dois inteiros X_s e Y_s , identificando a célula de saída da sala (X_s, Y_s). As demais linhas contém as posições (i,j) onde os armários estão posicionados. A entrada $i=-1$ e $j=-1$ indicam que não existem mais armários a serem alocados.

Saída

Seu programa deve produzir o caminho a ser percorrido (células (i,j)) e um número inteiro que indica o menor caminho (em número de células) da entrada até a saída da sala.

Exemplo:

| Entrada | Saída |
|---------|-------|
| 7 7 | 4 1 |
| 3 7 | 3 1 |
| 5 1 | 2 1 |
| 1 4 | 1 1 |
| 1 6 | 1 2 |
| 2 2 | 1 3 |

| | |
|-------|-----|
| 2 4 | 2 3 |
| 2 6 | 3 3 |
| 3 2 | 3 4 |
| 4 2 | 3 5 |
| 4 3 | 3 6 |
| 4 4 | 11 |
| 4 5 | |
| 4 6 | |
| 5 2 | |
| 5 6 | |
| 6 2 | |
| 6 4 | |
| 6 6 | |
| 7 4 | |
| -1 -1 | |

Instruções - Vídeo

A equipe deverá produzir um vídeo com no máximo 15 minutos, **e disponibilizá-lo em um link no drive.**

Além de apresentar o programa funcionando deverá responder no vídeo as questões a seguir:

- Se seu trabalho foi em equipe, explique como foi dividido. Quem fez o que/qual parte?
- Explique por meio de desenhos como ficaram suas estruturas de dados. Faça as justificativas necessárias em relação as suas escolhas de implementação.
- Qual foi a função mais difícil de implementar? Por quê? Quem implementou?
- Tem alguma funcionalidade solicitada no Projeto que não foi atendida?
- Tem algo que não funciona corretamente?
- Você testou para cenários diferentes?

- Saberá apresentar um cenário, diferente do apresentado como exemplo, que todas as saídas possíveis passam pelo mesmo número de células (“empate”)? Como seria a entrada?
- Saberá apresentar um cenário, diferente do apresentado como exemplo, que para escapar é preciso percorrer todas as células livres? Como seria a entrada?
- Se achar necessário apresente suas considerações sobre o Projeto.