

CEFET-MG Campus Leopoldina  
Engenharias: Computação e Controle e Automação  
Algoritmos e Estruturas de Dados  
TRABALHO PRÁTICO

Semestre: 2024.2  
Prof. Anderson Grandi Pires

Data de entrega: VER ABAIXO  
Local de entrega: SIGAA

## OBSERVAÇÕES

LEIA ATENTAMENTE AS INFORMAÇÕES ABAIXO

### Objetivos:

- Implementar um programa que empregue algumas estruturas de dados estudadas na disciplina, tais como fila e pilha.
- Empregar arquivos como fonte de entrada para problemas em geral.

### Observações:

- O Trabalho Prático será avaliado em **10 pontos**.
- A **linguagem C deverá ser utilizada** para implementar a atividade. Utilize as estruturas de dados desenvolvidas na disciplina para representar os dados do problema.
- O programa **deve funcionar** com labirintos **maiores**, desde que sejam definidos por matrizes  $N \times N$  ( $7 \leq N \leq 100$ ) e utilizem os caracteres '#' (parede) e '.' (célula livre). A descrição dos labirintos deverá ser obtida a partir de arquivos-texto.
- Seu programa **deve apresentar** uma saída semelhante àquela apresentada no **Exemplo de execução** disponível no fim deste documento.
- Insira seu nome completo em cada arquivo que compõe o projeto, na forma de um comentário geral (início do arquivo).
- **O programa será testado com labirintos de vários tamanhos**. A entrada e a saída sempre estarão nas células indicadas no enunciado.
- O trabalho será entregue em etapas, porém **a cada etapa deve ser entregue o trabalho completo** até aquela etapa, ou seja, na etapa 2 deve-se entregar o trabalho contendo o que foi desenvolvido nas etapas 1 e 2 e, assim, sucessivamente.

### ALGUMAS PENALIDADES NA CORREÇÃO

- A identificação de **cópia** resultará em penalidade de 10 pts.
- Etapa com **erro de compilação**: penalidade igual ao valor da etapa.
- Programa **desorganizado** e com problemas na **indentação**: penalidade de 20% do valor da etapa.

## TAREFAS A SEREM AVALIADAS

### ETAPA 1: 2 pontos

**Entrega: 10/02/2025**

[2 pontos] Desenvolver um programa para gerar, automaticamente, labirintos no padrão esperado como entrada. O programa deverá criar um arquivo denominado "labirinto.txt" com a descrição de um labirinto. Um labirinto deverá ser representado por uma matriz  $N \times N$ , sendo  $7 \leq N \leq 100$ . O arquivo possuirá os caracteres '#' e '.', onde '#' representa uma célula obstruída (parte de uma parede) e o caractere '.' representa uma célula vazia (espaço livre). O programa deverá ler da entrada o valor de N e um valor inteiro P, sendo  $10 \leq P \leq 80$ . O valor de P será usado na geração do labirinto e indica a probabilidade ( $P/100$ ) de uma célula ser do tipo parede. A borda externa do labirinto deve ser composta por paredes, independente do valor de P.

### ETAPA 2: 2 pontos

**Entrega: 12/02/2025**

- a) [1 ponto] Processar o arquivo "labirinto.txt" com a descrição de um labirinto gerado na ETAPA 1 (qualquer arquivo que possua um conteúdo compatível com a descrição de um labirinto apresentada na Figura 1-a deve funcionar). Um labirinto deverá ser representado por uma matriz  $N \times N$ , sendo  $7 \leq N \leq 1000$ .
- b) [1 ponto] Criar e inicializar uma matriz para representar o labirinto, conforme o conteúdo do arquivo e as informações apresentadas na Figura 1-c.

### ETAPA 3: 3 pontos

**Entrega: 14/02/2025**

- c) [1,5 ponto] Criar uma função para executar o passo de anotação, conforme descrito no enunciado.
- d) [1 ponto] Apresentar na tela (e em arquivos) as informações referentes às três partes da Figura 2, ou seja: Figura 2-a, Figura 2-b e Figura 2-c. A Figura 2-a deverá ser apresentada substituindo os caracteres '.' por espaços ' ' (conforme primeira figura do exemplo de execução abaixo). Gerar três arquivos F2a.txt, F2b.txt e F2c.txt com as mesmas informações que foram impressas na tela.

### ETAPA 4: 3 pontos

**Entrega: 16/02/2025**

- e) [1,5 ponto] Criar uma função para implementar as funcionalidades de extração do menor caminho.
- f) [1 ponto] Apresentar na tela (e em arquivos) as informações referentes ao melhor caminho, ou seja, apresentar a Figura 3-b, caso exista um caminho. Não havendo um caminho, apresentar na tela tal informação. O arquivo criado deverá ser nomeado F3b.txt
- g) [1 ponto] Apresentar a Figura 3-b incluindo cores: marcar os caracteres que representam a entrada e a saída na cor verde e o caminho (os 'X') na cor vermelha.

**Observação:** Todas as etapas compreendem um único programa que deverá solicitar ao usuário os valores de N e P, e apresentará como resultado da execução as figuras descritas a seguir.

## Descrição das Etapas do Trabalho Prático – Melhor (menor) caminho em um labirinto

### ETAPA 1

Geração de um arquivo-texto contendo a descrição de um labirinto com tamanho  $N \times N$  ( $7 \leq N \leq 100$ ). O valor de  $N$  deverá ser solicitado ao usuário. Além do tamanho, o usuário deverá fornecer um valor  $P$  ( $10 \leq P \leq 80$ ) que será usado para definir a densidade de paredes no labirinto, ou seja, quanto maior o valor de  $P$ , maior a probabilidade ( $P/100$ ) de existirem mais paredes no labirinto. O arquivo deverá conter somente os caracteres '#' e '.' sem espaços em branco entre eles. Os espaços, entre os caracteres, que aparecem nas figuras abaixo foram incluídos somente na impressão do labirinto, de modo que ele ficasse com boa apresentação na tela.

### ETAPA 2

Para o desenvolvimento desta etapa, os dados do labirinto deverão ser obtidos a partir do arquivo, sem acesso aos valores de  $N$  e  $P$  fornecidos pelo usuário. Assim, dado um arquivo-texto com a descrição de um labirinto cujo tamanho deverá ser obtido dos dados do arquivo, desenvolva um programa para processar o arquivo, armazenar as informações em uma matriz com valores 0 e -1 (Figura 1-c) e apresentar na tela a representação do labirinto, ou seja, as células livres e as células ocupadas (paredes). O arquivo-texto é composto dos caracteres '.' e '#' (Figura 1-a). O caractere '.' representa uma célula livre onde um agente virtual poderá se movimentar, enquanto o caractere '#' representa uma célula bloqueada, ou seja, pode ser considerada como parte de uma parede. Um agente virtual não poderá se movimentar para uma célula bloqueada. A movimentação possível do agente será para uma célula livre adjacente à posição atual, ou seja, para cima, para baixo, para esquerda ou para direita (Figura 1-d). As figuras abaixo descrevem: (a) um exemplo do conteúdo do arquivo "labirinto.txt", (b) a representação do labirinto referente àquele arquivo, (c) a matriz a ser criada e inicializada a partir das informações do arquivo e das demais informações disponibilizadas neste documento e (d) as regras de movimentação do agente virtual (Figura 1-d).

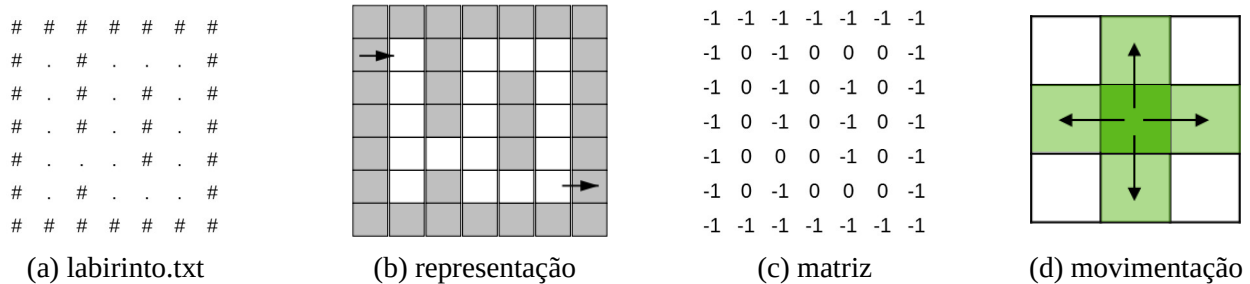


Figura 1. Matrizes bidimensionais representando um labirinto e a regra de movimentação de um agente.

O labirinto será representado por uma matriz quadrada  $L[n, n]$ , onde:  $L[i, j] = 0$  indica que a posição  $(i, j)$  está livre; e  $L[i, j] = -1$  indica que a posição  $(i, j)$  está obstruída por uma parede. Assumimos que a entrada do labirinto está na célula  $(1, 0)$  e que sua saída está na célula  $(n-2, n-1)$ , como indicado na Figura 1-b.

### ETAPA 3

Uma vez que a etapa anterior foi concluída, o próximo passo é efetuar a etapa de **anotação**, conforme descrita abaixo:

**Anotação:** Para encontrar um menor caminho, primeiro anotamos na matriz  $L$  o número mínimo de passos necessários para atingir cada uma das posições do labirinto, a partir da entrada. Assim, após a fase de anotação,  $L[i, j] = k$  indica que são necessários  $k$  passos para alcançar a posição  $(i, j)$ , como exemplificado na Figura 2-c. Caso uma posição qualquer da matriz permaneça com valor 0, isso significa que não existe um caminho que leve da entrada até essa posição.

---

#### Algoritmo Anotar(L)

---

$L[1,1] \leftarrow 1$

insira a posição  $(1, 1)$  em uma fila F vazia

**enquanto** F não estiver vazia **faça**

    remova uma posição  $(i, j)$  da fila F

**para** cada posição  $(x, y)$  adjacente a  $(i, j)$  tal que  $L[x, y] = 0$  **faça**

$L[x, y] \leftarrow L[i, j] + 1$

        insira a posição  $(x, y)$  na fila F

---

# # # # # # #	-1 -1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1 -1
# . # . . . #	-1 0 -1 0 0 0 -1	-1 1 -1 9 10 11 -1
# . # . # . #	-1 0 -1 0 -1 0 -1	-1 2 -1 8 -1 12 -1
# . # . # . #	-1 0 -1 0 -1 0 -1	-1 3 -1 7 -1 11 -1
# . . . # . #	-1 0 0 0 -1 0 -1	-1 4 5 6 -1 10 -1
# . # . . . #	-1 0 -1 0 0 0 -1	-1 5 -1 7 8 9 -1
# # # # # # #	-1 -1 -1 -1 -1 -1 -1	-1 -1 -1 -1 -1 -1 -1
(a) labirinto.txt	(b) representação	(c) anotação

Figura 2. Representações de um labirinto em (a) e (b). Resultado da etapa de anotação em (c).

#### ETAPA 4

A última etapa compreende a extração do menor caminho (caso exista) para ir da entrada até a saída do labirinto (Figura 3-b). O algoritmo para efetuar esta etapa está descrito a seguir:

##### Algoritmo Extrair(L)

---

*se*  $L[n-2, n-1] = 0$  **então** não existe um caminho, **pare** a execução da função  
*insira* a posição  $(n-1, n-1)$  em uma pilha P vazia  
**enquanto** a posição  $(1, 1)$  não estiver no topo da pilha P **faça**  
    *seja*  $(i, j)$  a posição existente no topo da pilha  
    *encontre* uma posição  $(x, y)$  adjacente a  $(i, j)$  tal que  $L[x, y] = L[i, j] - 1$   
    *insira* a posição  $(x, y)$  na pilha P  
*mostre* o caminho armazenado na pilha P

---

# # # # # # #	# # # # # # #
# . # . . . #	# X # # # #
# . # . # . #	# X # # # #
# . # . # . #	# X # # # #
# . . . # . #	# X X X # #
# . # . . . #	# # X X X #
# # # # # # #	# # # # # # #
(a) exemplo de labirinto	(b) melhor caminho

Figura 3. Conteúdo do arquivo de entrada com a representação de um labirinto (a) e um melhor caminho identificado como resultado da etapa de extração (b).

#### EXEMPLO DE EXECUÇÃO

(Neste exemplo **NÃO** estão **TODAS** as figuras solicitadas)

#	#	#	#	#	#	#
#		#				#
#		#		#		#
#		#		#		#
#				#		#
#		#				#
#	#	#	#	#	#	#

Labirinto

#	#	#	#	#	#	#
#	1	#	9	10	11	#
#	2	#	8	#	12	#
#	3	#	7	#	11	#
#	4	5	6	#	10	#
#	5	#	7	8	9	#
#	#	#	#	#	#	#

Etapa de anotação

#	#	#	#	#	#	#
#	X	#				#
#	X	#		#		#
#	X	#		#		#
#	X	X	X	#		#
#		#	X	X	X	#
#	#	#	#	#	#	#

Um menor caminho