

Quarto Exercício-Programa

Caça à cenoura

Entrega: 15 de novembro



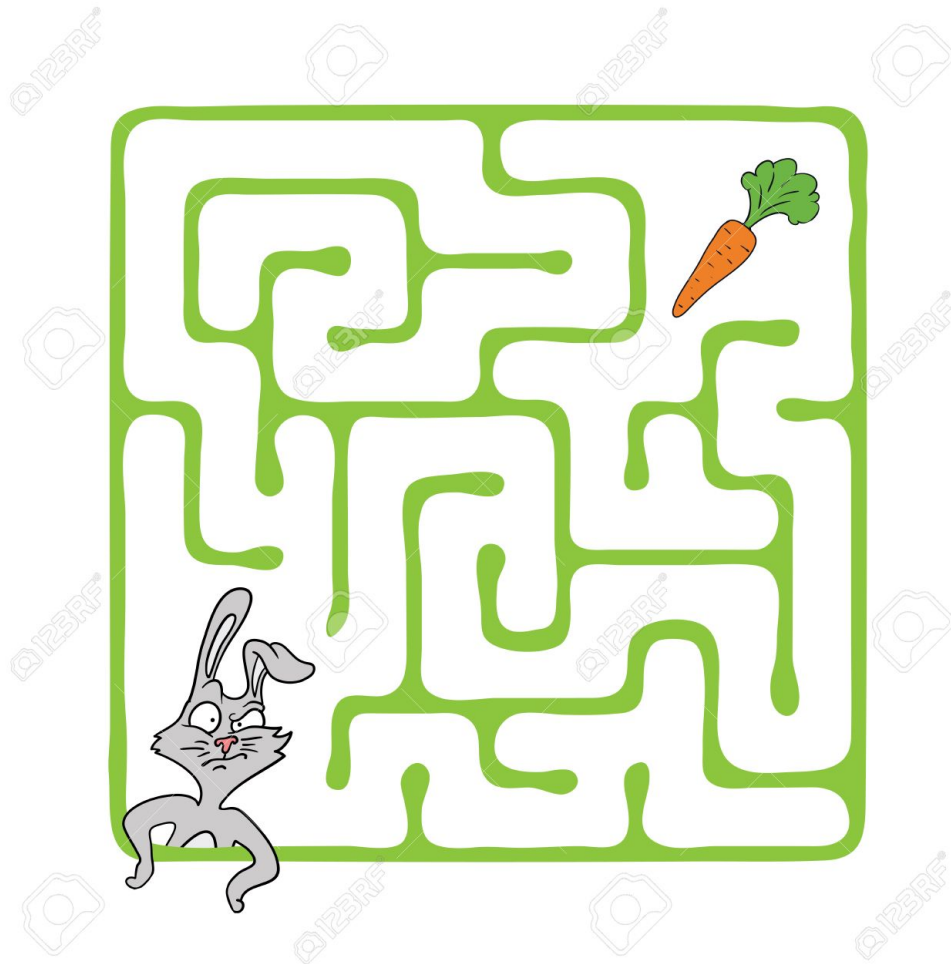
O objetivo deste exercício-programa é exercitar o uso de filas e cálculo de distâncias.

A cidade de Carrotsville tem como mascote um coelho que se chama Herbert. A região é um grande centro produtor de cenouras. Para chamar a atenção dos turistas para essa atividade econômica importante, a cidade oferece uma série de atrações inusitadas, como o cenoura boat, que funciona no lago da cidade, o sorvete de cenoura, chips de cenoura, camisetas temáticas, etc.

Uma das atrações mais concorridas é um labirinto na praça principal onde a prefeitura diariamente espalha algumas cenouras e, ao meio-dia, solta o Herbert para que ele saia em busca do seu almoço. Os turistas se divertem apostando quais das cenouras o Herbert vai almoçar.

O Herbert é um coelho preguiçoso. Ele se satisfaz com uma cenoura por refeição, e fica feliz quando caminha o mínimo possível até chegar ao seu almoço.

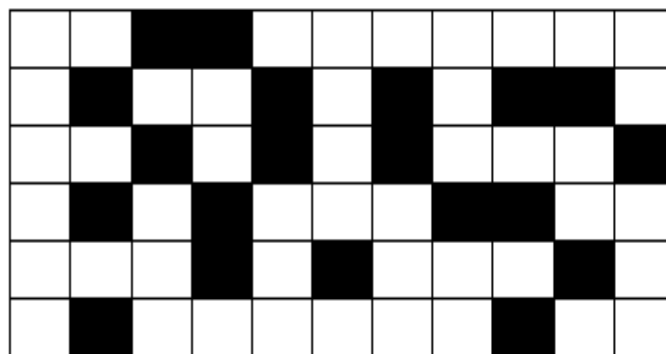
O seu objetivo é ajudar o Herbert a encontrar a sua cenoura do dia no menor número de passos possível.



Fonte: https://www.123rf.com/photo_31117852_stock-vector-vector-maze-labyrinth-game-for-children-with-rabbit-and-carrot-.html

Labirintos

Um labirinto será representado por dois inteiros positivos m e n , e uma matriz L binária com m linhas e n colunas. Uma posição com 0 é uma posição livre do labirinto, e uma posição com 1 é uma parede do labirinto. Por exemplo, o seguinte labirinto



seria representado por $m=6$ e $n=11$, e pela matriz

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Arquivos de entrada

O programa lerá a maior parte da informação de um arquivo texto indicado pelo usuário. Cada arquivo contém na sua primeira linha dois números, indicando os valores de m e n , e nas m linhas seguintes, contém uma sequência de n 0's ou 1's, com a linha correspondente da matriz que representa o labirinto.

Depois da matriz do labirinto, o arquivo contém numa linha um número inteiro positivo que indica quantas cenouras foram colocadas no labirinto, e em seguida, uma por linha, as posições das cenouras. Segue um exemplo de um tal arquivo, para o labirinto acima, com três cenouras:

```
6 11
0 0 1 1 0 0 0 0 0 0 0
0 1 0 0 1 0 1 0 1 1 0
0 0 1 0 1 0 1 0 0 0 1
0 1 0 1 0 0 0 1 1 0 0
0 0 0 1 0 1 0 0 0 1 0
0 1 0 0 0 0 0 0 1 0 0
3
1 11
3 4
4 10
```

Entrada e saída do programa

O programa deve pedir para o usuário digitar o nome de um arquivo onde se encontra a descrição de um labirinto e as posições das cenouras conforme o formato descrito na seção anterior.

Depois de ler do arquivo o labirinto e a posição das cenouras, o programa deve mostrar o labirinto para o usuário e pedir que ele digite a posição inicial do Herbert.

A partir daí, o programa deve calcular um caminho mínimo do Herbert até uma cenoura, e deve ao final imprimir novamente o labirinto, identificando a posição inicial do Herbert, das cenouras, e destacando o caminho determinado, e o comprimento deste caminho. Por exemplo, para o arquivo acima, se o usuário desse como posição inicial do Herbert a posição 1 1, seu programa poderia imprimir

0 Herbert achou uma cenoura em 20 passos!

```
H 0 1 1 0 * * * * * C
* 1 0 0 1 * 1 0 1 1 0
* 0 1 C 1 * 1 0 0 0 1
* 1 0 1 * * 0 1 1 C 0
* * * 1 * 1 0 0 0 1 0
0 1 * * * 0 0 0 1 0 0
```

O programa deve então perguntar ao usuário se ele quer dar uma nova posição inicial do Herbert, e repetir o processo até que o usuário digite que quer terminar.

Exemplo de saída do programa

```
*****
Carrotsville search engine
*****
```

0: carregar um novo labirinto e posição inicial do Herbert
 1: dar nova posicao inicial do Herbert no mesmo labirinto
 2: sair do programa

Digite a opcao desejada: 0

Digite o nome do arquivo com o labirinto: labirinto1.txt

Labirinto:

```
0 0 1 1 0 0 0 0 0 0
0 1 0 0 1 0 1 0 1 1
0 0 1 0 1 0 1 0 0 1
0 1 0 1 0 0 0 1 1 0
0 0 0 1 0 1 0 0 1 0
0 1 0 0 0 0 0 0 1 0
```

Digite a posicao inicial do Herbert: 1 1

0 Herbert achou uma cenoura em 20 passos!

```
H 0 1 1 0 * * * * * C
* 1 0 0 1 * 1 0 1 1 0
* 0 1 C 1 * 1 0 0 0 1
* 1 0 1 * * 0 1 1 C 0
* * * 1 * 1 0 0 0 1 0
0 1 * * * 0 0 0 1 0 0
```

0: carregar um novo labirinto e posição inicial do Herbert
 1: dar nova posicao inicial do Herbert no mesmo labirinto
 2: sair do programa

Digite a opcao desejada: 1

Digite a posicao inicial do Herbert: 6 10

0 Herbert achou uma cenoura em 4 passos!

```
0 0 1 1 0 0 0 0 0 0 C
0 1 0 0 1 0 1 0 1 1 0
0 0 1 C 1 0 1 0 0 0 1
0 1 0 1 0 0 0 1 1 C *
0 0 0 1 0 1 0 0 0 1 *
0 1 0 0 0 0 0 0 1 H *
```

0: carregar um novo labirinto e posição inicial do Herbert
 1: dar nova posicao inicial do Herbert no mesmo labirinto
 2: sair do programa

Digite a opcao desejada: 0

Digite o nome do arquivo com o labirinto: labirinto2.txt

Labirinto:

```
0 0 1 1 0
0 1 0 0 1
0 0 1 0 1
0 1 0 1 0
0 0 0 1 0
```

Digite a posicao inicial do Herbert: 1 1

0 Herbert não achou nenhuma cenoura!!! :(

```
H 0 1 1 0
0 1 0 0 1
0 0 1 C 1
0 1 0 1 0
0 0 0 1 0
```

Digite a opção desejada: 2

Tchau, Herbert!

Recomendações

Siga as [informações sobre os EPs](#), disponibilizadas no início do semestre.

Modularize o seu programa: pense em usar alguma das bibliotecas que desenvolvemos nas aulas.

Escreva funções para fazer cada tarefa diferente do EP: por exemplo, escreva uma função que imprime o menu, uma que faz a leitura do nome do arquivo e do labirinto, uma função que imprime o labirinto, etc.

Divirta-se com o EP!
