



Trabalho 02 - Estruturas de Controle, Estruturas de Repetição, Vetores, Matrizes e Cadeias de Caracteres

O trabalho está dividido em duas partes, sendo a primeira referente a um jogo de Caça-palavras e a segunda um jogo de Labirinto. A descrição de cada parte está a seguir:

OBS: O trabalho pode ser feito em grupos de no máximo 3 (três) pessoas, sendo obrigatório enviar ao professor por e-mail o nome dos membros e colocá-los como comentário no início do algoritmo. Apenas um membro do grupo deve enviar o trabalho ao fim. O aluno pode participar de apenas 1 (um) grupo.

1. **Caça-palavras:** é um jogo que consiste em uma matriz quadrada ou retangular de letras organizadas de maneira aleatória com a presença de algumas palavras em sua forma escrita correta ou invertida. O objetivo é encontrar todas as palavras completas presentes na matriz. Desta forma, dado um caça-palavras composto por uma matriz de caracteres, em que o usuário deve informar o tamanho (número de linhas e número de colunas) e os caracteres de cada posição da matriz, desenvolver um programa que busque na matriz a ocorrência de uma palavra informada pelo usuário. Caso a palavra seja encontrada o programa deve informar em que coordenadas encontrou e se esta em ordem normal ou invertida de escrita.

As palavras podem estar distribuídas pela matriz na horizontal e vertical, em ordem normal ou inversa. Para encerrar o jogo o usuário deve digitar a palavra **sair**, caso contrário deve continuar pedindo uma nova palavra para buscar.

Um exemplo do resultado esperado pode ser observado na Figura 1.

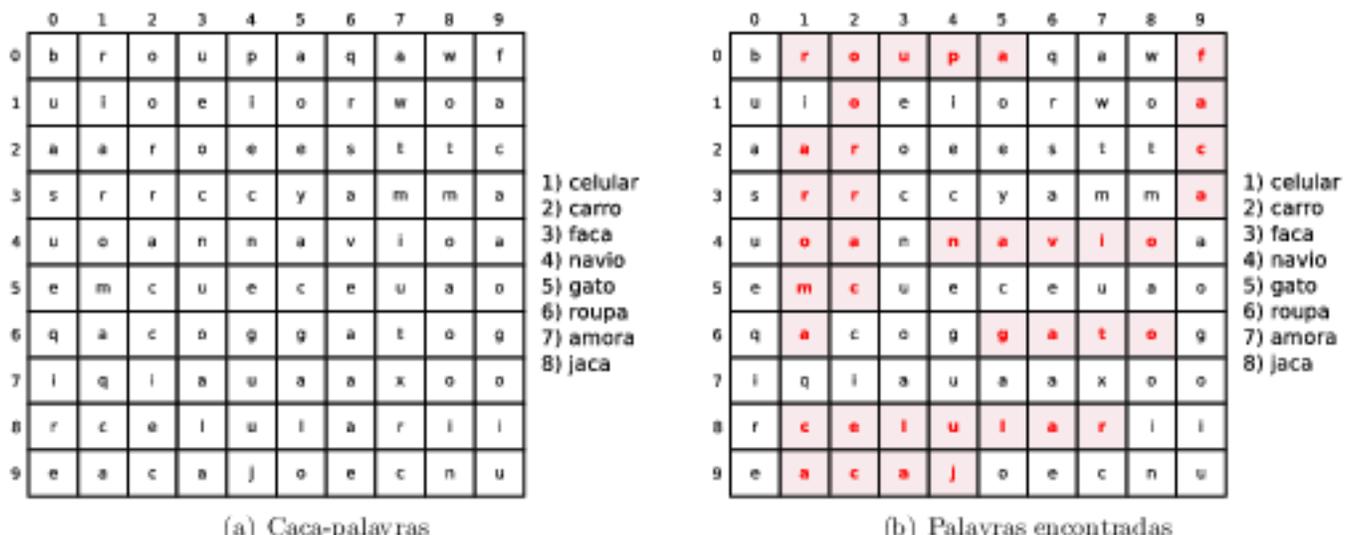


Figura 1: Exemplo de um caça-palavras.

2. Cego no Labirinto de Vidro: o jogo do labirinto corresponde a um conjunto de caminhos com o propósito de atraparlar o jogador na busca por um único caminho que o guie para fora de um ambiente. Os labirintos podem conter uma ou várias saídas e um ou vários possíveis caminhos. No jogo proposto neste trabalho um cego é posicionado em uma determinada coordenada de uma grade/matriz e a partir desta posição deve seguir certas direções para tentar encontrar uma saída. Porém, o chão deste labirinto é de vidro, e a cada passo a posição anterior é quebrada impossibilitando o retorno.

O cego do jogo somente pode andar seguindo uma ordem de escolhas, sendo elas: primeiro ele tenta andar ao norte, após tenta nordeste, leste, sudeste, sul, sudoeste, oeste e por fim noroeste. Caso ele não consiga andar em nenhuma direção ele “morre”. Lembre que o caminho do nosso personagem pode conter obstáculos como paredes e quadros de vidro quebrados por já ter passado por aqueles locais. A(s) saída(s) do jogo esta(ao) posicionada(s) na(s) lateral(is) do quadro.

O jogo deve receber como entrada o tamanho da matriz (linhas e colunas) e todos os valores de suas posições (cada uma delas). Em seguida, deve receber a posição inicial do cego (número da linha e número da coluna) e a partir daí o jogo deve tomar as decisões dos passos sozinho independente do jogador, exibindo na tela para qual posição o cego andou. O jogo termina quando o cego consegue chegar a uma saída ou quando não existem opções de passos possíveis.

Se o personagem chegar a uma saída você deve imprimir na tela que ele saiu e em qual posição, caso contrário imprimir que ele morreu e em qual posição. A formatação de saída deve seguir o exemplo dado:

Solução positiva

```
-> Andou para: [1] [12]
-> Andou para: [0] [12]
-> Andou para: [0] [13]
Saiu na posicao [0] [13]
```

Solução negativa

```
-> Andou para: [1] [7]
-> Andou para: [0] [7]
-> Andou para: [0] [8]
Morreu na posicao [0] [8]
```

Nas figuras 2 e 3 são apresentados exemplos de quadros e posição iniciais do cego e as decisões tomadas pelo jogo.

	0	1	2	3	4	5
0	0	0	0	2	0	0
1	2	0	0	2	0	2
2	0	0	2	2	0	0
3	2	0	0	2	0	0
4	2	2	0	2	2	0
5	0	0	0	0	0	0

N	N	N
N	⊗	N
N	N	N

(a) Disposição inicial.

	0	1	2	3	4	5
0	0	0	0	2	3	0
1	2	0	0	2	0	2
2	0	0	2	2	0	3
3	2	0	0	2	0	0
4	2	2	0	2	2	0
5	0	0	0	0	0	0

N	N	N
N	⊗	S
N	N	N

(b) Posição das saídas (verde), do jogador (vermelho) e das paredes (marrom). Em Azul as possíveis direções.

	0	1	2	3	4	5
0	0	0	0	2	3	0
1	2	0	0	2	0	2
2	0	0	2	2	0	3
3	2	0	0	2	0	0
4	2	2	0	2	2	0
5	1	0	0	0	0	0

N	N	S
N	⊗	S
N	N	N

(c) O programa decide pelo primeiro possível passo, o sentido nordeste.

	0	1	2	3	4	5
0	0	0	0	2	3	0
1	2	0	0	2	0	2
2	0	0	2	2	0	3
3	2	0	0	2	0	0
4	2	2	0	2	2	0
5	1	1	0	0	0	0

S	S	N
N	⊗	N
N	S	S

(d) A primeira posição segundo a ordem de direções é o norte.

	0	1	2	3	4	5
0	0	0	0	2	3	0
1	2	0	0	2	0	2
2	0	0	2	2	0	3
3	2	0	0	2	0	0
4	2	2	1	2	2	0
5	1	1	0	0	0	0

S	N	N
S	⊗	N
N	N	N

(e) O próximo passo é para o oeste pois para o sul já estava quebrado.

	0	1	2	3	4	5
0	0	0	0	2	3	0
1	2	0	0	2	0	2
2	0	0	2	2	0	3
3	2	0	1	2	0	0
4	2	2	1	2	2	0
5	1	1	0	0	0	0

S	S	N
N	⊗	N
N	N	N

(f) Seguindo o próximo é ao norte.

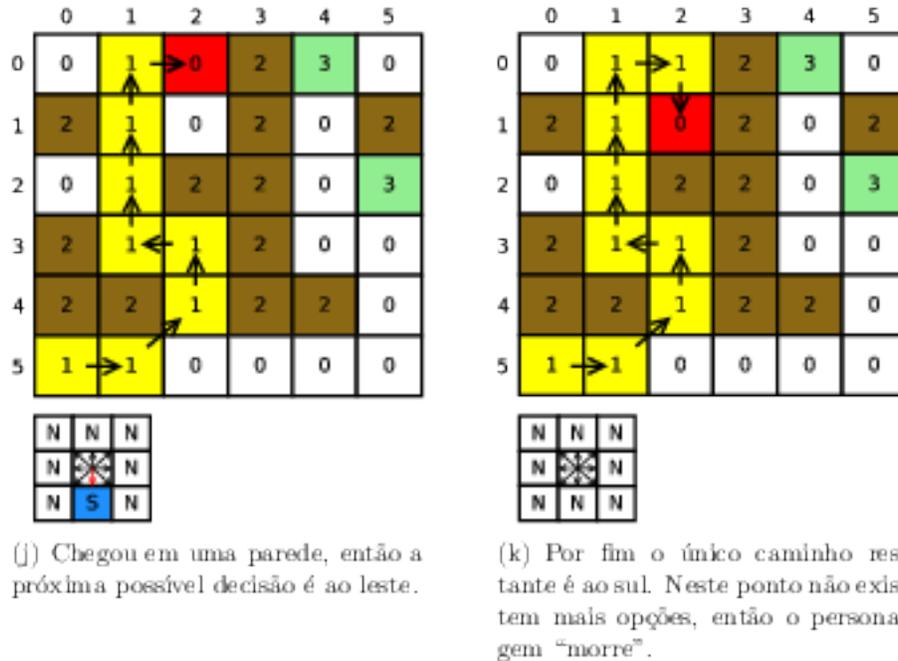
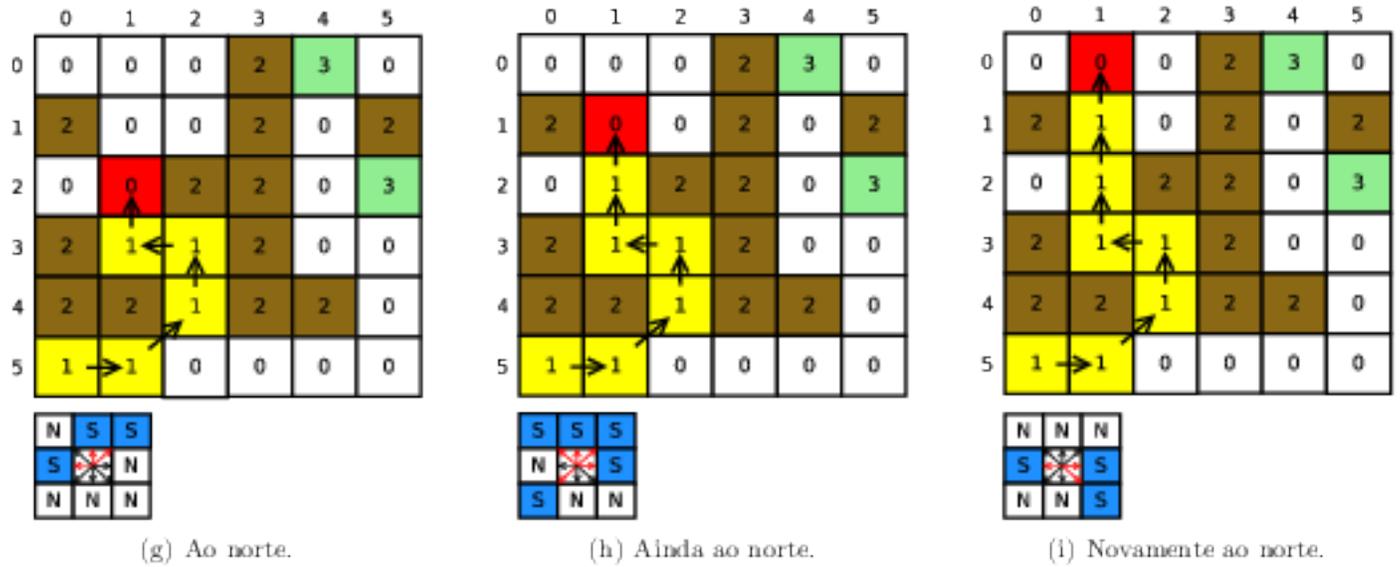
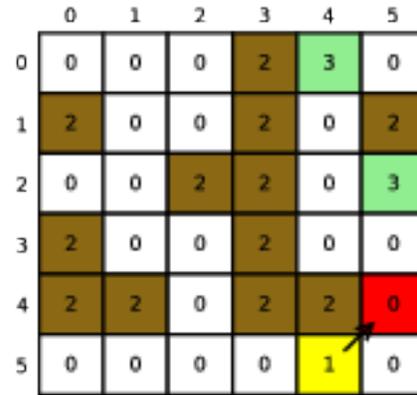


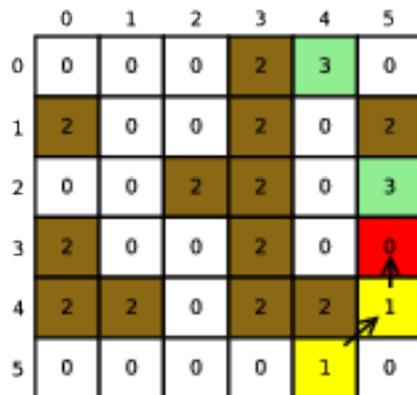
Figura 2: Exemplo de caminho sem saída (negativo).



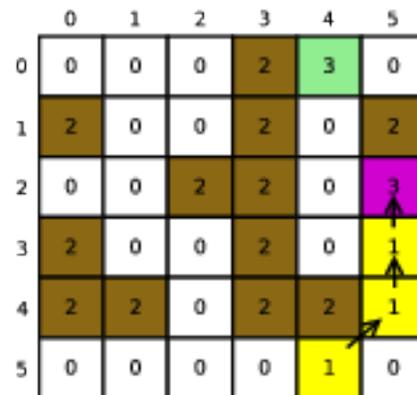
(a) O personagem inicia em coordenadas diferentes.



(b) O primeiro passo é ao nordeste.



(c) O próximo é ao norte.



(d) No passo seguinte o personagem caminha ao norte e encontra uma saída, terminando o programa.

Figura 3: Exemplo de caminho com saída (positivo).