

lista 07

Questão 1 (Pilha de posições). Considere o seguinte fragmento de código que define uma classe que é uma pilha de posições. Complete os métodos `empilhar(Posicao x)`, `Posicao desempilhar()` e `int vazia()` e execute o código

```
#include <stdio>

class Posicao{
    public:
        int l;
        int c;
};

class PilhaPosicao {
    public:
        Posicao* v;
        int tam_max;
        int topo;
        int cont;    //número de elementos na pilha

        // método construtor, inicializa a pilha
        PilhaPosicao(int tamanho){
            v = new Posicao[tamanho];
            tam_max = tamanho;
            topo = -1;
            cont = 0
        }

        // método empilhar
        void empilhar(Posicao p){
            ...
        }

        // método desempilhar
        Posicao desempilhar(){
            ...
        }

        // método pilha vazia
        int vazia(){
            ...
        }
};
```

Questão 2 (Labirinto). Considere uma matriz M de dimensões $n \times n$ preenchida com os caracteres '#', 'e', 's' e espaço em branco.

$$\begin{bmatrix} \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ e & & & & & & & & & \# \\ \# & \# & \# & & \# & \# & \# & & & \# \\ \# & \# & \# & & \# & \# & & & & \# \\ \# & & & & \# & \# & \# & \# & & \# \\ \# & \# & \# & & \# & \# & \# & \# & & \# \\ \# & \# & & & & & & & & \# \\ \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \# & & & & & & & & & s \\ \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \end{bmatrix}$$

Escreva um programa que recebe um labirinto e checa se há um caminho entre 'e' e 's'.

$$\begin{bmatrix} \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & & & \# \\ \# & \# & \# & & \# & \bullet & \# & \# & & \# \\ \# & \# & \# & & \# & \bullet & \# & & & \# \\ \# & \bullet & \bullet & \bullet & \# & \bullet & \# & \# & \# & \# \\ \# & \bullet & \# & \bullet & \# & \bullet & \# & \# & \# & \# \\ \# & \bullet & \# & \bullet & \bullet & \bullet & & & & \# \\ \# & \bullet & \# & \# & \# & \# & \# & \# & \# & \# \\ \# & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \circ \\ \# & \# & \# & \# & \# & \# & \# & \# & \# & \# \end{bmatrix}$$

Caso haja, seu algoritmo deve retornar o caminho.