

Algoritmos e Estrutura de Dados

Trabalho prático

Orientações: Este trabalho pode ser desenvolvido individualmente ou em dupla. A entrega deverá ser realizada por uma única pessoa, no sistema run.codes (<https://run.codes>), até o prazo de 12/12/2022. Para isso, faça seu cadastro no sistema e procure pela disciplina “90000002 - Algoritmos e programação de computadores”. Para participar da disciplina, será solicitado um código:

- Turma 2022/1: o código é E371;

Não serão aceitos trabalhos indicados como plágio pelo sistema run.codes. No cabeçalho de seu código-fonte devem conter os nomes dos integrantes do grupo.

1 Descrição

Neste trabalho, escreva um programa que simula um caça-palavras. Esse programa deve receber uma matriz quadrada, formada por caracteres (letras ou números), e uma palavra. Deve determinar em quais posições essa palavra se encontra na matriz. Uma palavra está em uma matriz se sua seqüência de letras pode ser encontrada na matriz, de tal maneira que:

- a primeira letra da palavra existe na matriz;
- o restante da palavra está disposto em uma reta que se inicia na posição da primeira letra e segue em um dos oito sentidos diferentes (N, S, L, O, NE, NO, SE, SO);
- ao atingir a borda, a busca deve continuar na borda oposta.

Entrada: Esse programa deve receber a dimensão n da matriz. Em seguida, deve ler n strings de tamanho n . Cada string representa uma linha da matriz. Após a leitura das n strings, será informada a palavra a ser buscada na matriz. Não é necessária nenhuma verificação. Como exemplo, considere a entrada:

```
10
VTOWAIAUBA
JXAESSUFSR
DEBIAICALI
CEBMHIZQZR
ASACRTBLMC
SVGTBCNZON
ACFEZKULCA
AL7ICASADZ
ECNYVLXLWQ
CEBHCIZCJK
CASA
```

Nessa entrada, o primeiro valor, 10, representa a dimensão da matriz (10×10). Em seguida, são informadas 10 strings de tamanho 10, que formarão a matriz, e, após a leitura dessas strings, tem-se a palavra CASA, que será buscada na matriz. A partir dessa entrada, tem-se a matriz disponível na tabela (1).

Saída: Como saída, seu programa deve imprimir a posição inicial em que a palavra foi encontrada na matriz (caso exista alguma) seguida de sua posição final. A busca deve ser realizada percorrendo primeiramente as colunas depois as linhas. Como exemplo, na matriz da tabela (1), ao buscar a palavra CASA, seu programa deve imprimir:

V	T	O	W	A	I	A	U	B	A
J	X	A	E	S	S	U	F	S	R
D	E	B	I	A	I	C	A	L	I
C	E	B	M	H	I	Z	Q	Z	R
A	S	A	C	R	T	B	L	M	C
S	V	G	T	B	G	N	Z	O	N
A	C	F	E	Z	K	U	L	C	A
A	L	7	I	C	A	S	A	D	Z
E	C	N	Y	V	L	X	L	W	Q
C	E	B	H	C	I	Z	C	J	K

Tabela 1: Matriz de entrada.

(3,0)(6,0)
(4,3)(4,0)
(4,9)(4,2)
(7,4)(7,7)
(9,0)(2,7)
(9,4)(2,4)
(9,7)(2,4)

em que cada elemento $(x_i, y_i)(x_f, y_f)$ representa os pontos iniciais (x_i, y_i) e finais (x_f, y_f) , formados pelas linhas e colunas, em que passam as retas. As palavras encontradas estão em negrito na matriz da tabela (1).

2 Material a ser Entregue

O código-fonte de sua solução deverá ser entregue pelo sistema **run.codes**. Para isso, utilize o código de matrícula E371. No sistema run.codes o seu código será compilado e testado, a fim de avaliar seu correto funcionamento.

Você deve enviar o código-fonte chamado **caca_palavras.c**. O código-fonte, bem como a correta execução do programa equivale a 60% da nota. 40% da nota serão calculados com base na organização, na legibilidade e na eficiência do seu código.

1. Legibilidade: O código é fácil de ser lido e entendido ou foi escrito de uma forma que dificulta sua compreensão? Sugestões: utilize nomes de funções e variáveis que descrevam suas utilidades ou funcionalidades. Quando necessário, comente o código, explicando trechos que não são óbvios.
2. Organização e reaproveitamento de código: o código está organizado e dividido em pequenas partes? Sugestões: deve-se utilizar funções para melhorar a organização do código e evitar repetição desnecessária de trechos de código.
3. Eficácia e eficiência: as operações são realizadas de forma correta e da melhor forma possível? Sugestões: para verificar a eficácia de seu programa, estarão disponíveis casos de testes no sistema run.codes. Quanto à eficiência, evite usar variáveis desnecessárias, ou tipos de variáveis que são além do que o necessário, ou mesmo criar funções que não serão utilizadas.

3 Conclusão

Será atribuído zero aos seguintes casos:

- Código-fonte não compilar.
- Nenhum teste for executado com sucesso.
- Plágio.

Bom trabalho.