Desafio de Fundamentos de Programação
"Caça-palavras"

Universidade Tecnológica Federal do Paraná (UTFPR), campus Apucarana
Curso de Engenharia de Computação
Disciplina de Fundamentos de Programação - FD61A
Prof. Dr. Luiz Fernando Carvalho
Prof. Dr. Rafael Gomes Mantovani

---

1 Objetivo

O caça-palavras é um passa-tempo popular cujo objetivo é encontrar certas palavras em um diagrama. Neste desafio, sua tarefa é implementar um problema que resolve um caça-palavras. As palavras estão escondidas em uma matriz de letras e podem estar na vertical, horizontal, ou diagonal. E, além disso, as palavras podem aparecer tanto em ordem direta quanto na ordem invertida.

---

2 Entrada

A entrada do programa é feita via arquivo texto. Ela consiste de:

- uma linha com dois inteiros `m` e `n` que correspondem ao número de linhas e de colunas da matriz de letras embaralhadas;
- as `m` linhas seguintes contém `n` letras cada uma, separadas por um espaço simples;
- a linha imediatamente após a matriz contém um número `p` de palavras a serem procuradas;
- as `p` linhas seguintes contêm uma palavra em cada linha.

```
3 8
v e d j n a e o
i p y t h o n u
s u e w e t a e
1
python
```

Figura 1: Exemplo de entrada para o resolvedor de caça-palavras.

```
. . . . . . . .
. p y t h o n .
. . . . . . . .
```

Figura 2: Exemplo de saída mostrando as palavras encontradas pelo resolvedor de caça-palavras.

3 Saída

Como saída, imprima no console, e também em um arquivo texto a ser gerado, uma matriz modificada mostrando as palavras encontradas. Essa saída é bem parecida com a matriz de entrada, mas contém o caractere '.' nas posições em que não foi encontrada uma palavra. Um exemplo de saída para a entrada da Figura 2 é mostrado na figura abaixo.
Dica: Quando forem executar o programa com os arquivos texto é necessário manipular os argumentos **argc** e **argv** da função main. Para isso, deve-se executar o programa por linha de comando, obedecendo o seguinte padrão:

<nome do programa> <arquivo de entrada> <arquivo de saída>

Por exemplo, se o programa fonte se chamar "cacapalavras.c", o comando será:

cacapalavras entrada.txt saida.txt

Dentro da função `main`, o valor de `argc` indica o número de parâmetros recebidos para a execução. Como são inseridos apenas os nomes dos arquivo de entrada e saída, nesse caso temos `argc` = 3 (os dois arquivos mais o nome do programa). Além disso:

- `argv[0]` contém o nome do programa;
- `argv[1]` o nome do arquivo de entrada; e
- `argv[2]` o nome do arquivo de saída.

4 O que deve ser entregue?

Para entrega da atividade, o aluno deve enviar ao professor:

a) todos os arquivos desenvolvidos em C para a resolução do problema;

b) relatório individual descrevendo a modelagem realizada, explicando todas as tomadas de decisão (estruturas de dados usadas, lógica desenvolvida, etc.). Constar no relatório também casos de teste realizados (pode inserir figuras para mostrar as execuções realizadas);

c) apresentação oral para o professor em horário a ser combinado.

---

5 Orientações gerais

Além da funcionalidade desejada, implementar também o controle de erros, para lidar com exceções que possam ocorrer, como, por exemplo:

- problemas nas aberturas dos arquivos de entrada e saída;
- arquivo de entrada vazio (sem informação);
- arquivo de entrada fora do padrão esperado (dados faltantes, quantidade errada de linhas informadas, quantidade errada de palavras que se deseja buscar, etc.);
- etc.

Para acompanhamento do desenvolvimento, pode-se criar um repositório individual com o código desenvolvido na página pessoal do aluno no `github`. Ao elaborar seu programa, crie um único arquivo fonte (.c) seguindo o padrão de nome especificado:

FDP-AT01-CacaPalavras-<NomeAluno>.c

Exemplo:

`FDP-AT01-CacaPalavras-RafaelMantovani.c`

A entrega da atividade será via Moodle: o link será disponibilizado na página da disciplina. O prazo para a entrega é o final do semestre letivo corrente.

---

6 Links úteis

Números aleatórios em C:

- http://linguagemc.com.br/valores-aleatorios-em-c-com-a-funcao-rand/
- https://www.ime.usp.br/~pf/algoritmos/aulas/random.html
- http://www.cplusplus.com/reference/cstdlib/rand/

Arquivos em C:

- https://www.inf.pucrs.br/~pinho/LaproI/Arquivos/Arquivos.htm
- https://www.geeksforgeeks.org/basics-file-handling-c/
- https://www.programiz.com/c-programming/c-file-input-output

Argumentos de linha de comando (argc e argv):

- https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm
- http://linguagemc.com.br/argumentos-em-linha-de-comando/
- http://www.univasf.edu.br/~marcelo.linder/arquivos_pc/aulas/aula19.pdf
- http://www.inf.ufpr.br/cursos/ci067/Docs/NotasAula/notas-31_Argumentos_linha_comando.html
- http://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node145.html

---

Referências

[1] Thomas H. Cormen,; Ronald Rivest; Charles E. Leiserson; Clifford Stein. Algoritmos - Teoria e Prática - 3ª Ed. Elsevier - Campus, 2012.

[2] Nivio Ziviani. Projeto de algoritmos com implementações: em Pascal e C. Pioneira, 1999.

[3] André Backes. Linguagem C: completa e descomplicada. Rio de Janeiro, RJ: Elsevier, 2013. 371 p.

[4] Luis Damas. Linguagem C. 10a edição. Editora LTC. 2007.
