

SCC0201 - Introdução à Ciência de Computação II

Trabalho 2

Professor: Diego Raphael Amancio

Estagiária PAE:
Ana Caroline Medeiros Brito

DATA LIMITE: 30/11/2024 12hrs

Entrega no `run.codes`

Dúvidas:
`carol_mb@usp.br`

Soletra no terminal

Concursos de soletração são uma clássica competição onde os participantes devem soletrar palavras específicas com diferentes níveis de dificuldade. Recentemente, variações dessa competição vem sendo difundidas online em formato de jogo como o caso disponível em <https://www.nytimes.com/puzzles/spelling-bee> para o inglês e <https://g1.globo.com/jogos/soletra/> para o português.

Nesse jogo é esperado que o usuário liste todas as variações de palavras possíveis que atendam ao critério da configuração inicial: são válidas palavras que sejam compostas por um conjunto de 7 letras, sendo uma letra indicada como requisito para todas as palavras. Se as letras dadas como configuração inicial são P (obrigatória), A, T, O, E, F e I, são palavras válidas **PATO, PATA e APTO**, são palavras *inválidas* **TATO e AFETO**.

Tarefa

Sua tarefa é implementar um soletra para o terminal que tenha os seguintes comandos:

- `inicio` [lista com 7 letras separadas por espaço]: recebe como entrada 7 letras, sendo a primeira a letra aquela que deve estar presente em todas as palavras
- `palavra` [entrada]: recebe uma palavra que será validada, isto é, se atende aos critérios como uma resposta válida. Se for uma resposta válida, será impressa a mensagem “sucesso + 1”, caso contrário, será impressa a mensagem “palavra invalida”. Caso o usuário tenha terminado de listar todas as palavras válidas, então a mensagem “parabens! voce encontrou todas as palavras” deve ser mostrada e o programa é encerrado
- `progresso`: lista a quantidade de palavras identificadas e faltantes, agrupadas por número de letras
- `solucao`: a lista com todas as palavras que ainda não foram identificadas é impressa (agrupadas por número de letras) e o programa é encerrado

Um arquivo auxiliar com a lista de palavras válidas será disponibilizado e também parte do projeto no `run.codes`. Consideramos válidas apenas palavras com 4 ou mais letras. Todas as entradas serão em letra maiúscula.

Entrada

A entrada é composta por várias linhas, espere sempre que o primeiro comando será a inicialização do jogo.

Exemplo:

```
inicio P A T O E F I  
palavra PATO  
palavra PATA  
palavra APTO  
palavra TATO  
palavra AFETO  
progresso  
solucao
```

Saída

Exemplo:

```
sucesso + 1
sucesso + 1
sucesso + 1
palavra invalida
palavra invalida
progresso atual:
(4 letras) 3 palavra(s) encontrada(s) / 28
palavra(s) faltando
(5 letras) 0 palavra(s) encontrada(s) / 33
palavra(s) faltando
(6 letras) 0 palavra(s) encontrada(s) / 22
palavra(s) faltando
(7 letras) 0 palavra(s) encontrada(s) / 7
palavra(s) faltando
(8 letras) 0 palavra(s) encontrada(s) / 3
palavra(s) faltando
para encerrar o jogo estavam faltando as
palavras:
(4 letras) APEA, APTA, APTO, OPIO, OPOE, OPTA,
OPTE, OPTO, PAIO, PAPA, PAPE, PAPO, PATA,
PATO, PEAO, PIAI, PIAO, PIEI, PIFA, PIFE,
PIFO, PIPA, PIPI, POTE, TAPA, TAPE, TAPO,
TIPO, TOPA, TOPE, TOPO
(5 letras) APEAI, APEEI, APEIA, APEIE, APEIO,
APITA, APITE, APITO, APOIA, APOIE, APOIO,
ETAPA, ETIPE, FIAPO, OPTAI, OPTEI, PAPAI,
PAPAO, PAPEI, PATIO, PEITA, PEITE, PEITO,
PIFAI, PIFEI, POETA, TAIPA, TAPAI, TAPAO,
TAPEA, TAPEI, TOPAI, TOPEI
(6 letras) APATIA, APITAI, APITEI, APOIAI,
APOIEI, ETIOPE, PAPAIA, PATETA, PATIFE,
PATOTA, PEITAI, PEITEI, PEPITA, PIPETA,
TAPEAI, TAPEEI, TAPEIA, TAPEIE, TAPEIO,
TAPETE, TIPOIA, TOPETE
(7 letras) APETITE, ATAPETA, ATAPETE, ATAPETO,
EPITETO, EPOPEIA, ETIOPIA
(8 letras) ATAPETAI, ATAPETEI, EPITAFIO
fim!
```

Exigências

- O arquivo `valid_words.txt` deve ser utilizado como base para identificar quais palavras são válidas para uma dada configuração inicial.
- Pelo menos o algoritmo de busca binária deve ser implementado.
- Arquivo README informando resumidamente quais estruturas de dados e algoritmos foram implementados e quais as motivações para essas escolhas.

Entrega

- A solução deve ser desenvolvida em C. Verifique se o programa compila sem erros ou mensagens de alerta.
- A entrega será realizada pelo **run.codes**, será possível verificar se seu programa está se comportando de forma esperada através dos casos de testes disponíveis.
- Trata-se de um trabalho em grupo, basta que apenas um membro envie a solução no sistema.
- Pode assumir que não serão dadas entradas mal formatadas, por exemplo, letras minúsculas ou não inicialização como primeiro comando.

Critérios de avaliação

O projeto será avaliado principalmente levando em consideração:

- Processamento correto de entradas e saídas do programa;
- Realização das tarefas descritas;
- Atender as exigências especificadas;
- Bom uso das técnicas de programação.