Tarefa 3 - Homônimos e parentes

Publicado em 19/08/2021

MC202

VOLTAR TOPO

Prazo de entrega recomendado: ***29/08/2021***

Você buscará nomes de estudantes homônimos e parentes em uma turma. Para isso, você deve ler, armazenar, organizar e manipular strings.

O professor do curso de Estruturas de Dados recebeu a lista de estudantes. Como ele gosta de uma aula bastante participativa, ele gostaria de chamar cada estudante pelo nome, mas para não se confundir, é necessário identificar os estudantes com nomes homônimos. Por exemplo, *Antonio Silva* e *Antonio Pereira* são homônimos.

Além disso, é muito importante que estudantes conversem e estudem juntos. Por esse motivo, em determinadas atividades, os estudantes serão separados em grupos. Para tornar as atividades mais interessantes e enriquecedoras, ele gostaria de que os membros de cada grupo não se conhecessem. Assim, o professor também precisa descobrir quais estudantes são da mesma família. São considerados da mesma família estudantes que têm o mesmo último sobrenome. Por exemplo, *Maria Santos* e *Luiz Santos* são parentes.

Sua tarefa é escrever um programa `classificar.c` que recebe a lista de estudantes e identifica ou homônimos ou parentes.

Entrada

A primeira linha contém um inteiro n indicando o número de estudantes ($0 < n \leq 100$). A segunda linha contém uma palavra `homonimos` ou `parentes` indicando o que o professor deseja fazer. As n linhas seguintes contêm os nomes dos estudantes, separados por `_` e com no máximo 30 caracteres cada.

Cada estudante só tem um nome e um sobrenome, por exemplo `Luiz_Silva`, você não encontrará casos como `Luiz_da_Silva`.

Exemplo de entrada

```
13
homonimos
Luiz_Silva
Sandra_Lima
Paulo_Pereira
Francisca_Borges
Marcelo_Barbosa
Manoel_Araujo
Geraldo_Ramos
Carlos_Barbosa
Luis_Costa
Francisca_Cardoso
Juliana_Santana
Ada_Ramos
Manoel_Carvalho
```

Saída

A saída consiste da lista dos nomes de estudantes que são homônimos ou parentes, a depender da opção selecionada.

Exemplo de saída 1

Quando a opção selecionada for `homonimos`, então a saída deverá ser ordenada lexicograficamente.

```
Francisca_Borges
Francisca_Cardoso
Manoel_Araujo
Manoel_Carvalho
```

Exemplo de saída 2

Se opção selecionada tivesse sido `parentes`, então a saída deveria ser ordenada por sobrenome e, em seguida, lexicograficamente.

```
Carlos_Barbosa
Marcelo_Barbosa
Ada_Ramos
Geraldo_Ramos
```

Critérios

Para ler um nome completo, utilize a função `scanf` passando a especificação de formato `%s`. Se desejar, você pode utilizar as funções `strlen`, `strcpy` e `strcmp` da biblioteca `<string.h>`; outras funções são proibidas. Há diversas unidades de teste, contendo casos de teste em que:

- os nomes da entrada já estão em ordem e é solicitado listar homônimos;
- os nomes da entrada não estão ordenados e é solicitado listar homônimos;
- os nomes da entrada não estão ordenados e é solicitado listar parentes.

Pense com calma, escreva seus algoritmos antes de programar e resolva os conjuntos com casos de teste mais simples primeiro. Para passar na tarefa é obrigatório resolver corretamente pelo menos os casos de teste com entradas ordenadas em que é solicitado listar homônimos. Os casos de teste em que é solicitado listar parentes são opcionais e **não** irão influenciar na nota, mas tente fazê-los depois.

Correção

Esta tarefa será corrigida automaticamente sempre que você realizar um `git push`. Depois de terminada a tarefa, deve-se utilizar o botão na interface de notas para solicitar a correção de um monitor.

Turma AB: O peso desta tarefa será 1.

Turma E: Você deverá apresentar esta tarefa a um monitor PED. Para isso, você deve procurar atendimento em algum horário com monitor PED e digitar ***apresentar 3*** no canal `fila-apresentar`.