

MC202EF - Estruturas de Dados

Lab 08

(O conteúdo necessário para realizar este laboratório vai até a unidade:
Tabela de Espalhamento)

20 de outubro de 2016

Jogo da Memória

1. Problema¹

Um emissora de televisão projetou um novo jogo da memória para as tardes de domingo; o problema é que ela ainda não sabe como encontrar as respostas rapidamente, atrapalhando a transmissão ao vivo. Neste jogo, os participantes recebem um texto que deve ser lido em um tempo pré-determinado. Após o tempo se esgotar, é exibida uma frase de uma ou mais palavras em um telão e cada participante deve escrever a palavra que ele acredita que aparece mais vezes imediatamente depois da frase fornecida. Com isso, se um jogador acertar uma palavra X que vem depois da frase, ele **ganha** uma pontuação dada por

$$100 \times (\text{número de vezes que } X \text{ vem logo depois da frase}),$$

Já se ele escrever uma palavra que não aparece logo depois da frase, mas pelo menos um outro jogador tiver acertado uma palavra, ele **perde** a pontuação dada por:

$$10 \times (\text{número de palavras que vem logo depois da frase}),$$

contando as repetições (se uma palavra aparecer várias vezes depois da frase). Ninguém fica negativo, se um jogador perder mais pontos do que tiver, ele volta para zero.

Calculados os pontos de cada jogador, uma nova rodada é iniciada com uma nova frase. A cada rodada, os pontos dos jogadores são acumulados.

Para resolver o problema da emissora, você precisa desenvolver um programa que calcula a pontuação de cada jogador ao longo das rodadas. De modo a testar seu programa, foram fornecidos textos quaisquer, dos quais são extraídas as sequências de palavras e palpites de jogadores fictícios ao longo de diversas rodadas.

¹ Esse problema foi adaptado do original gentilmente cedido por Guilherme Pollo.

2. Entrada

A primeira linha contém o número n de palavras que formam o texto seguinte. Todas as palavras têm comprimento de no máximo 48 caracteres (sem pontuação ou acentuação). Após as n palavras, há dois inteiros: r e i , que indicam, respectivamente, os números de rodadas e de participantes. Cada uma das r linhas seguintes contém um inteiro positivo k seguido de k palavras, compondo a frase atual. Após essa linha, há i linhas indicando o palpite (uma única palavra) de cada jogador, em ordem.

```
49
when mortals are alive they worry about death
when theyre full they worry about hunger
theirs is the great uncertainty
but sages dont consider the past
and they dont worry about the future
nor do they cling to the present
and from moment to moment they follow the way
3 3
1 they
cling
worry
dont
2 the past
and
and
now
1 about
me
you
they
```

3. Saída

Cada linha corresponde a uma rodada, contendo o número da rodada e as pontuações acumuladas de todos os jogadores, em ordem, como no exemplo:

Rodada	1:	100	200	100
Rodada	2:	200	300	90
Rodada	3:	200	300	90

Na primeira rodada todos jogadores escolheram palavras que aparecem logo depois de “they”. Entretanto, o segundo participante optou pela palavra “worry”, que aparece 2 vezes entre um total de 5 palavras (“worry”, “worry”, “dont”, “cling”, “follow”) que seguem imediatamente “they” – logo sua pontuação foi de 100×2 . Na segunda rodada, os dois primeiros jogadores optaram pela mesma palavra, que é a única ocorrência imediata após “the past” no texto e ganharam 100×1 pontos; já o terceiro jogador escolheu uma palavra que não aparece logo depois da frase e perdeu 10×1 pontos. Na última rodada, ninguém acertou a palavra e, portanto, ninguém perde ponto.

4. Dicas

- a. Lembrem-se de liberar o espaço após o uso.
- b. Você pode usar algo como `printf("Rodada %2d: %4d %4d %4d\n",)` para obter uma saída com números alinhados.
- c. Pesquise boas funções de hashing para strings.

5. Avaliação

5.1 A nota desse laboratório será calculada da seguinte maneira:

- 7 pontos proporcionais ao número de casos de teste corretos;
- 3 pontos referentes à qualidade de código: legibilidade, algoritmo, memory leak, boas práticas de programação (comentários no código, escolha do nomes para variáveis, reutilização de funções que possam melhorar a apresentação do código, etc...).

5.2 Critérios avaliados neste laboratório:

Além de passar nos casos de teste, é obrigatório:

- ter como base para sua implementação tabelas de espalhamento.

6. Entrega

A submissão de código deve ser feita no Run.codes em no máximo 10 tentativas até o dia 18/11/16 às 23:59:59.