

Primeiro Trabalho - Terceira Etapa

MC102 - Algoritmos e Programação de Computadores

Instituto de Computação
Universidade Estadual de Campinas

1º semestre de 2012

Turmas Q e R

Professor: André Santanchè

PED: Daniel Cintra Cugler

PAD: Luis Henrique Pauleti Mendes

Parte A

Resumo

Um museu de artes resolveu fazer uma apuração de qual de suas seções o público tem mais gostado. Para isto colocou uma urna na saída do museu, onde o visitante escreve em um pedaço de papel o nome da seção do museu que mais gostou e coloca na urna. Para a apuração dos dados no final do mês, um funcionário reúne todos os pedaços de papel e os organiza em ordem alfabética.

Escreva um programa que solicite estes dados pelo teclado e imprima ao final da digitação o nome das duas seções que receberam mais votos. A entrada de dados será feita em ordem alfabética e cada item digitado corresponde ao nome escrito em um pedaço de papel.

Não se sabe no início do programa quantas seções existem e isto não deve ser solicitado ao usuário. Quando o usuário digitar "FIM" ao invés do nome da seção (flag) significa que ele terminou de entrar todos os dados.

Detalhamento

Considere que não há limite de seções diferentes que podem ser digitadas. O programa deve ser capaz de lidar com qualquer número de seções.

Se houver empate - ou seja, três seções tiverem o mesmo número de votos (duas delas ocupando o primeiro e segundo lugares), ou duas seções tiverem o mesmo número de votos ocupando o segundo lugar - será dado prioridade à seção que apareceu primeiro na lista.

Entrada

- Sequência de entradas separadas por ENTER.
- O programa pode ter 1 até n entradas, cada uma em um scanf próprio. O flag "FIM" será exclusivamente todo em caixa alta.

Saída

- Deve ser apresentado no console:
 - primeiro lugar: <nome>; segundo lugar: <nome>
- No lugar de <nome> aparecem o nome do primeiro e segundo lugar respectivamente.
- Se por alguma razão não houver segundo lugar, não deve aparecer a mensagem correspondente nem o ponto-e-vírgula. Se não houver primeiro lugar nada deve ser impresso no console.

Entrega

- A entrega será feita até o dia 03/04/2012 através do Susy.

Parte B

Resumo

Estenda o programa feito na primeira etapa de modo que ele otimize programas de compras que são feitas em diversos planetas, de modo que se faça o mínimo de viagens possíveis.

Detalhamento

Tal como na segunda fase, utilize o *framework* mercante para ler pelo teclado uma string contendo um programa do robô e executar as instruções.

Nesta versão serão usadas apenas instruções de compra e viagem. Entretanto, antes de executar o programa, agrupe as compras por planeta de tal modo que seja realizado o menor número de viagens possíveis.

Por exemplo, na sequência:

```
viaja-Tander;indica-braco;compra-5;viaja-Bantor;indica-cabeca;compra-7;viaja-Tander;indica-perna;compra-2
```

Podem ser agrupadas todas as compras de Tander para que sejam realizadas de uma só vez antes de ir para Bantor.

Além disso, rearrume as viagens dos planetas de modo que elas sejam feitas em ordem crescente do número do planeta. No exemplo dado, a viagem a Bantor (4) seria feita antes de Tander (5). As compras devem ser feitas por ordem crescente do número da mercadoria. Não execute viagem para planetas em que não serão compradas mercadorias.

Não é necessário alterar a string que contém os comandos, apenas a execução dos mesmos deve ser agrupada e ordenada. Considere que sempre haverá saldo suficiente para se realizar todas as compras.

Entrada

- Sequência de instruções separadas por ponto-e-vírgula.
- Instruções que serão usadas:
 - `viaja-<número_planeta>`
 - `viaja-<nome_planeta>`
 - `indica-<número_mercadoria>`
 - `indica-<nome_mercadoria>`
 - `compra-<quantidade>`

Saída

O programa a ser desenvolvido não deve imprimir nada na tela, apenas as saídas emitidas pelo *framework* devem aparecer.

Entrega

- A entrega será feita até o dia 09/04/2012 através do Susy.