

Universidade Federal do Espírito Santo
Centro Tecnológico
Programação II - 2013/1
Prof. Flávio Varejão
2º Trabalho Computacional

Balancete de Loja de Departamentos

Ao final de cada ano uma empresa composta por uma rede de lojas de departamentos faz um balancete a respeito das vendas de seus produtos. Para facilitar a execução do balancete, a empresa decidiu te contratar para fazer um programa em C que analise os dados das compras efetuadas por clientes ao longo do ano e produza alguns resultados informativos. O programa terá de ler de um arquivo texto de entrada chamado “entrada.txt” os dados do balancete, que se apresentam no seguinte formato:

<código da loja>
<código do departamento>
<código do produto> <preço unitário> <taxa de lucro>
<código do cliente>
<quantidades compradas separadas por branco>
<posições i, j, k e l de elementos em listas>

Os dados do balancete estão organizados por código de loja, departamento e de turma. Assim, em cada loja existem vários departamentos; cada departamento vende vários produtos; cada produto é comprado por vários clientes em uma ou mais ocasiões. Junto com o código do produto também é informado o preço unitário do produto e a taxa de lucro obtida pela empresa no total faturado com as vendas daquele produto. O número inteiro -1 é usado como indicador de fim de uma seção de informação. Assim, o número -1 aparece para indicar o fim das compras de um cliente, o fim da lista de clientes que compraram um certo produto, o fim da lista de produtos de um determinado departamento, o fim da lista de vendas dos departamentos de uma loja, o fim dos dados de todos os departamentos de uma loja e o fim dos dados de toda a rede de lojas da empresa.

A última linha do arquivo “entrada.txt” conterá quatro números inteiros separados por espaço (i, j, k e l). Cada um desses números indicará uma posição em uma das listas que serão geradas pelo sistema (considerar que a primeira posição de cada lista é representada pelo número 1).

Apresenta-se a seguir um exemplo do arquivo “entrada.txt”

Exemplo do arquivo “entrada.txt”:

```
123
2628
13457 304.50 10.0
101356
1 1 3 5 -1
209876
8 2 4 -1
130999
1 2 -1
-1
27777 45.32 5.0
```

329876
85 3 -1
-1
-1
2611
77277 5.12 7.5
999876
8 33 -1
-1
-1
-1
989
1126
90909 3.2.3 3.2
222222
800 233 -1
-1
-1
-1
-1
1 2 1 3

Todos dados são números inteiros, com exceção de preços unitários e taxas de lucro, que são números em ponto flutuante. Não se sabe a priori o número de lojas, o número de departamentos por lojas, o número de produtos vendidos em um departamento, o número de clientes que compraram um determinado produto, nem o número de ocasiões que um dado cliente comprou um determinado produto. Como dito anteriormente, o número -1 é usado como indicador para permitir determinar essas informações na medida que os dados são lidos.

ATENTE para o fato que agora um mesmo cliente pode comprar diferentes produtos em diferentes departamentos de diferentes lojas. ATENTE também para o fato de que um mesmo produto pode ser vendido em diferentes lojas e que o seu preço unitário pode variar de loja para loja.

Considerando que não existem mais do que mil clientes e mil produtos, seu programa deve ler iterativamente cada linha da entrada (obedecendo ao formato descrito acima) e obter as seguintes informações:

1. A lista de clientes ordenados decrescentemente pelo valor total de compras que efetuaram.
2. A lista de clientes ordenados crescentemente pela quantidade de compras que efetuaram.
3. A lista de produtos ordenados crescentemente pelo valor total de suas vendas.
4. A lista de produtos ordenados decrescentemente pelo lucro que proporcionaram.

Os resultados serão salvos em um arquivo nomeado "saida.txt" no seguinte formato:

<soma>
<pares (código, valor de compras) da lista 1 separados por espaços>
<pares (código, quantidade de compras) da lista 2 separados por espaços>
<pares (código, valor de vendas) da lista 3 separados por espaços>
<pares (código, lucro) da lista 4 separados por espaços>

O componente <soma> corresponde a soma dos códigos do i-ésimo elemento da lista

1, do j-ésimo elemento da lista 2, do k-ésimo elemento da lista 3 e do l-ésimo elemento da lista 4

Um exemplo do arquivo “saida.txt” é fornecido a seguir:

```
288888
(111111, 30.00) (222222, 20.00) (333333, 10.00)
(333333, 100) (111111, 200) (222222, 300)
(12345, 1000.00) (98765, 1200.00) (54321, 1500.00)
(98765, 100.00) (12345, 72.00) (54321, 35.00)
```

Data de Entrega: O trabalho deverá ser entregue até as 23:59 horas do dia 04/09/2013 (domingo).

Forma de Entrega:

1. Compacte o arquivo texto (o nome do arquivo DEVE ser trab2.c) com o código fonte do programa do seu trabalho e envie o arquivo compactado para o e-mail fvarejao@inf.ufes.br.
2. O assunto do e-mail deverá ser o seguinte (somente o que está entre aspas duplas): "\prog2:trab2:nome:". Substitua nome pelo seu primeiro nome e último sobrenome, separados por espaço. Compacte o arquivo fonte utilizando o programa (ou comando) zip e envie o arquivo compactado em anexo. Ressaltamos que o arquivo trab2.zip gerado deverá conter apenas o arquivo com o código fonte do seu trabalho (não podem ter arquivos executáveis ou qualquer outro arquivo).
3. Após o envio do trabalho você receberá uma confirmação de recebimento do trabalho. Caso isso não ocorra, reenvie o seu trabalho para o mesmo e-mail.
4. O recebimento dos trabalhos é automatizado. Siga as instruções à risca pois algum erro na submissão pode inviabilizar a entrega do seu trabalho. Não deixe para enviar seu trabalho nos momentos finais de seu prazo. É comum a ocorrência de problemas em virtude de erros na submissão. Logo, enviem com algumas horas de antecedência para que haja tempo hábil para eventuais correções.

Veja abaixo um exemplo de um e-mail de envio do trabalho:

Para: fvarejao@inf.ufes.br

De: Joao da Silva

Assunto: \prog2:trab2:Joao Silva:

Anexo: trab2.zip

ATENÇÃO:

- a. No assunto, a disciplina (prog2) e a identificação do trabalho (trab2) devem ser escritos todos em letras minúsculas.
- b. NÃO escreva o seu nome com caracteres estendidos (á, ã, etc).
- c. NÃO utilize contas de e-mail do BOL, HOTMAIL ou TERRA, pois as mensagens enviadas por esses servidores não seguem o padrão normal. Assim, o programa de recebimento automático não consegue receber o seu e-mail.

Outras Observações Importantes:

1. Os trabalhos serão verificados automaticamente por uma ferramenta de detecção de plágio. Em caso de detecção de cópia (parcial ou integral), todos os envolvidos recebem nota ZERO. Em outras palavras, tanto os alunos que copiaram quanto o que deixou copiar recebem ZERO.
2. Envie o trabalho no prazo especificado e no formato especificado. Trabalhos recebidos fora do prazo ou em formato inadequado recebem nota ZERO.
3. O trabalho deve ser enviado estritamente para os e-mails especificados acima.

4. Trabalho que não compila recebe nota ZERO. Não adianta nem submeter.
5. Os trabalhos serão compilados e verificados usando o compilador gcc no sistema operacional Linux.
6. Os programas serão avaliados pela sua correção durante a execução e também pelo estilo de programação. Serão observados particularmente se os programas possuem os comentários apropriados, se usam nomes significativos para as variáveis e funções, se o código está indentado corretamente e se utilizam modularização sempre que possível e apropriado.
7. Caso haja algum erro neste documento, serão divulgadas erratas. Portanto, fique atento às observações do professor durante as aulas e aos avisos na página do curso.

BOM TRABALHO !!!