

INSTITUTO SUPERIOR TÉCNICO

Introdução aos Algoritmos e Estruturas de Dados

2011/2012

Enunciado do 3^o Projecto

Data de entrega: 09 de Dezembro de 2011 às 23h59

1 Introdução

Neste projecto pretende-se desenvolver um programa que manipule informação relativa a produtos disponíveis e compras efectuadas num supermercado. Durante o dia são atendidos um conjunto de clientes e processados os respectivos carrinhos de compras. No final do dia, o programa deverá produzir uma listagem dos talões emitidos e estatísticas do movimento da caixa, nomeadamente número de clientes atendidos, total facturado e produto mais vendido.

2 Especificação do Programa

O programa recebe como *input* informação acerca dos produtos disponíveis e das compras efectuadas num supermercado (conjunto de carrinhos de compras).

Cada produto tem um código, um valor unitário sem IVA, uma taxa de IVA e uma descrição. Um carrinho de compras é composto por um conjunto de produtos.

Para cada carrinho de compras, o programa deverá apresentar no *output* o talão a emitir. Cada talão contém uma lista dos produtos comprados. Para cada produto é apresentada a taxa de IVA, o código, a descrição, o valor unitário com IVA, as unidades compradas e o preço total. Os produtos no talão são ordenados por ordem crescente da taxa de IVA e código do produto. No final do talão é indicado o valor total a pagar.

O programa deverá também apresentar no *output* o número de carrinhos de compras processados, o valor total das compras nos carrinhos e o código do produto mais vendido.

3 Dados de Entrada

O programa deverá ler os dados de entrada a partir do *standard input*. O formato será o seguinte:

- Uma sequência de linhas com informação relativa a produtos. Cada linha tem o seguinte formato:
 - Um carácter ‘p’ seguido de um espaço em branco;
 - Código do produto (sequência única de dez dígitos) seguido de um espaço em branco;
 - Preço unitário do produto sem IVA seguido de um espaço em branco;
 - IVA do produto (0.06, 0.13 ou 0.23) seguido de um espaço em branco;
 - Descrição do produto (sequência de caracteres com tamanho máximo 20);
- Uma sequência de carrinhos de compras com o seguinte formato:
 - Uma linha com o carácter ‘c’ que indica o início das compras;
 - Uma sequência de linhas com o código dos produtos no carrinho (um por linha);
 - Uma linha com o carácter ‘t’ que indica o fim das compras e escrita do talão.

Assuma que os dados de entrada para o programa não contêm erros sintácticos, isto é, obedecem sempre ao formato descrito nesta secção.

4 Dados de Saída

O programa deverá escrever no *standard output* a seguinte informação:

- Para cada carrinho de compras listar o respectivo talão com a seguinte informação:
 - Uma listagem dos produtos comprados ordenados por ordem crescente da taxa de IVA. Dentro de cada taxa os produtos deverão ser ordenados por ordem crescente do código. Para cada produto deve ser mostrado:
 - * Uma linha com a taxa de IVA (6%, 13% ou 23%), o código do produto e a respectiva descrição, separados por um espaço em branco;
 - * Uma linha com o número de unidades compradas, o carácter ‘x’, o preço unitário com IVA e o preço total, separados por um espaço em branco. Os preços são apresentados com duas casas decimais. O preço unitário com IVA deve ser arredondado antes do cálculo do preço total.
 - Valor total das compras.

- Após a escrita dos talões deve ser mostrada a seguinte informação:
 - Uma linha com o número de carrinhos de compras processados;
 - Uma linha com o valor facturado no conjunto dos carrinhos;
 - Uma linha com o código do produto mais comprado. Em caso de empate deve ser escrito o produto com código mais baixo. Esta linha deve ser omitida no caso do número de carrinhos ser zero.

5 Exemplo

5.1 Dados de Entrada

Suponha que o ficheiro de entrada, que designaremos por `in.txt`, tem o seguinte conteúdo:

```
p 0000000001 6.20 0.06 ananas dos acores
p 0000000002 5.50 0.23 biscoitos da beira
p 0000000003 0.80 0.06 pao alentejano
p 0000000004 4.50 0.23 doce amendoa algarve
p 0000000005 7.75 0.13 vinho do dao
p 0000000006 15.33 0.13 queijo da serra
c
0000000001
0000000002
0000000003
0000000004
0000000005
0000000006
t
c
0000000001
0000000006
0000000002
0000000001
0000000003
0000000003
0000000006
0000000004
0000000003
0000000006
t
```

5.2 Dados de Saída

Para o ficheiro de entrada `in.txt` descrito na secção anterior, o programa deverá escrever na saída a seguinte informação:

```
6% 0000000001 ananas dos acores
1 x 6.57 6.57
6% 0000000003 pao alentejano
1 x 0.85 0.85
13% 0000000005 vinho do dao
1 x 8.76 8.76
13% 0000000006 queijo da serra
1 x 17.32 17.32
23% 0000000002 biscoitos da beira
1 x 6.77 6.77
23% 0000000004 doce amendoa algarve
1 x 5.54 5.54
45.81
6% 0000000001 ananas dos acores
2 x 6.57 13.14
6% 0000000003 pao alentejano
3 x 0.85 2.55
13% 0000000006 queijo da serra
3 x 17.32 51.96
23% 0000000002 biscoitos da beira
1 x 6.77 6.77
23% 0000000004 doce amendoa algarve
1 x 5.54 5.54
79.96
2
125.77
0000000003
```

6 Compilação do Programa

O compilador a utilizar é o `gcc` com as seguintes opções de compilação: `-ansi -Wall -pedantic`. Para compilar o programa deve executar o seguinte comando:

```
$ gcc -ansi -Wall -pedantic -o proj3 *.c
```

o qual deve ter como resultado a geração do ficheiro executável `proj3`, caso não haja erros de compilação. A execução deste comando não deverá escrever qualquer resultado no terminal. Caso a execução deste comando escreva algum resultado no terminal, considera-se que o programa não compilou com sucesso. Por exemplo, durante a compilação do programa, o compilador não deve escrever mensagens de aviso (*warnings*).

7 Execução do Programa

O programa deve ser executado da forma seguinte:

```
$ ./proj3 < in.txt > out.txt
```

8 Entrega do Projecto

A entrega do projecto deverá respeitar o procedimento seguinte:

- Na página da disciplina aceda ao sistema para entrega de projectos. O sistema será activado uma semana antes da data limite de entrega. Instruções acerca da forma de acesso ao sistema serão oportunamente fornecidas.
- Efectue o upload de um ficheiro arquivo com extensão `.zip` que inclua os ficheiros fonte (`.c`) e cabeçalho (`.h`) que constituem o programa.

Para criar um ficheiro arquivo com a extensão `.zip` deve executar o seguinte comando na directoria onde se encontram os ficheiros com extensão `.c` e `.h`, criados durante o desenvolvimento do projecto:

```
$ zip proj3.zip *.c *.h
```

- Como resultado do processo de upload será informado se a resolução entregue apresenta a resposta esperada num conjunto de casos de teste.
- **O sistema não permite submissões com menos de 10 minutos de intervalo para o mesmo grupo.** Exemplos de casos de teste serão oportunamente fornecidos.

- Data limite de entrega do projecto: **23h59 do dia 9 de Dezembro de 2011**. Até à data limite poderá efectuar o número de entregas que desejar, sendo utilizada para efeitos de avaliação a **última** entrega efectuada. Deverá portanto verificar cuidadosamente que a última entrega realizada corresponde à versão do projecto que pretende que seja avaliada. Não serão abertas excepções.

9 Avaliação do Projecto

9.1 Componentes da Avaliação

Na avaliação do projecto serão consideradas as seguintes componentes:

1. A primeira componente avalia o desempenho da funcionalidade do programa realizado. Esta componente é avaliada entre 0 e 16 valores.
2. A segunda componente avalia a qualidade do código entregue, nomeadamente os seguintes aspectos: comentários, indentação, estruturação, modularidade, abstracção, entre outros. Esta componente poderá variar entre -4 valores e +4 valores relativamente à classificação calculada no item anterior e será atribuída na discussão final do projecto.

Nesta componente **será também utilizado o sistema valgrind for forma a detectar fugas de memória (“memory leaks”) ou outras incorrecções no código, que serão penalizadas**. Aconselha-se por isso que os alunos utilizem este sistema para fazer debugging do código e corrigir eventuais incorrecções antes da submissão do projecto.

Durante a discussão final do projecto será averiguada a participação de cada elemento do grupo na realização do projecto, bem como a sua compreensão do trabalho realizado, sendo a respectiva classificação ponderada em conformidade, isto é, elementos do mesmo grupo podem ter classificações diferentes. **Elementos do grupo que se verifique não terem participado na realização do respectivo projecto terão a classificação de 0 (zero) valores.**

9.2 Atribuição Automática da Classificação

A classificação da primeira componente da avaliação do projecto é obtida automaticamente através da execução de um conjunto de testes executados num computador com o sistema operativo **GNU/Linux**. Torna-se portanto essencial que o código compile correctamente e que respeite o formato de entrada e saída dos dados descrito anteriormente. Projectos que não obedeçam ao formato indicado no enunciado serão penalizados na avaliação automática, podendo, no limite, ter 0 (zero) valores se falharem todos os testes. Os testes considerados para efeitos de avaliação podem incluir (ou não) os disponibilizados na página da disciplina, além de um conjunto de testes adicionais. A execução de cada programa em cada teste é limitada na quantidade de memória que pode utilizar, até um máximo de 64 MBytes, e no tempo total disponível para execução, sendo o tempo limite distinto para cada teste.

Note-se que o facto de um projecto passar com sucesso o conjunto de testes disponibilizado na página da disciplina não implica que esse projecto esteja totalmente correcto. Apenas indica que passou alguns testes com sucesso, mas este conjunto de testes não é exaustivo. É da responsabilidade dos alunos garantir que o código produzido está correcto.

Em caso algum será disponibilizada qualquer tipo de informação sobre os casos de teste utilizados pelo sistema de avaliação automática. A totalidade de ficheiros de teste usados na avaliação do projecto serão disponibilizados na página da disciplina após a data de entrega.

9.3 Detecção de Cópias

A avaliação dos projectos inclui a utilização de um sistema para detecção de situações de cópia entre projectos. A submissão de um projecto pressupõe o compromisso de honra que o trabalho incluso foi realizado única e exclusivamente pelos alunos referenciados nos ficheiros submetidos para avaliação. A quebra deste compromisso, ou seja a tentativa de um grupo se apropriar de trabalho de outrem (sejam colegas ou outra pessoa), tem como consequência a reprovação de todos os alunos envolvidos (incluindo quem possibilitar a ocorrência de cópias) à disciplina de IAED.

Toda e qualquer situação de fraude ou facilitação de fraude terá então como consequência a reprovação imediata à disciplina de IAED neste semestre, assim como a comunicação da ocorrência ao respectivo Coordenador de curso e ao Conselho Pedagógico do IST para sanções adicionais de acordo com as regras aprovadas pela UTL e publicadas em Diário da República.

9.4 Considerações adicionais

Todos os programas são avaliados do modo seguinte:

```
$ ./proj3 < in.txt > out.txt; diff out.txt exp.txt
```

em que o ficheiro `exp.txt` representa o resultado esperado da execução do programa para os dados de entrada definidos pelo ficheiro `in.txt`. A impossibilidade de verificar automaticamente o resultado da execução de um dado programa implica uma penalização de **100%** na componente de avaliação automática. Considera-se que um programa passou um teste com sucesso se o resultado produzido por esse programa for exactamente igual ao resultado esperado, o que significa que o comando `diff` não deverá encontrar diferenças entre o resultado produzido pelo programa submetido e o esperado. Para poder ser avaliado, um projecto deverá compilar correctamente num computador com o sistema operativo **GNU/Linux**, sendo o utilizado o compilador **gcc** da **GNU**. A entrega de código não compilável, ou a não inclusão de qualquer dos ficheiros requeridos, ou a utilização de nomes diferentes para o ficheiro executável conduz a uma classificação de 0 (zero) valores. Não serão aceites quaisquer justificações.