MC202 — ESTRUTURAS DE DADOS

Lab 02 — Vetores Dinâmicos Ordenados

Problema

Uma *exchange* é uma plataforma de troca de criptomoedas. Para se comprar ou vender criptomoedas nessa plataforma, cada investidor cria uma ordem de compra ou de venda informando a quantidade de criptomoedas e preço unitário por criptomoedas e o sistema busca por compradores e vendedores que se encaixem em seu critério. Existem alguns tipos de ordens. Neste laboratório estamos interessados apenas em ordens casadas.

Ordens casadas são executadas com **preço igual** ao especificado pelo investidor. Por exemplo, no caso de uma ordem casada de venda, o preço da ordem de compra (presente na tabela de negociação) deve ser igual ao preço de venda. Cada ordem contém um identificador, a quantidade de criptomoedas e o preço desejado por criptomoedas. Quando existem duas ou mais ordens de venda ou de compra com **preços iguais na tabela de negociação**, o critério de desempate é o identificador: a ordem com o menor identificador é escolhida.

Quando uma ordem de compra ou de venda é aberta no sistema, deve-se tentar executá-la imediatamente. Caso a quantidade de criptomoedas da ordem de compra ou venda presente na tabela de negociação não seja suficiente para a quantidade de criptomoedas desejadas para a ordem criada, a transação ocorre e a quantidade de criptomoedas faltante é registrada na ordem aberta e essa vai para a tabela de negociação caso não exista outras ordens que completem a transação. Essa ordem fica na tabela de negociação aguardando por outras ordens que coincidam com seu critério.

Entrada

A entrada é composta por várias linhas, onde cada linha contém uma única ordem. Cada ordem recebe um identificador de forma sequencial, i.e., a primeira ordem recebe o identificador 1 (um), a segunda ordem recebe o identificador 2 (dois) e assim por diante. Cada ordem é composta por um caractere **`C`** ou **`V`** (onde **`C`** indica compra e **`V`** indica venda), um número inteiro positivo que indica a quantidade de criptomoedas desejadas e o preço desejado por criptomoeda com precisão de duas casas decimais. O fim da entrada ocorre quando for lido o caractere **`#`**.

Saída

O seu programa deve produzir uma ou mais linhas de saída sempre que ocorra alguma transação. Quando uma transação ocorre, a saída deve ser no formato "`ordem` ***`i`*** `vende para ordem` ***`j`*** `a quantidade` ***`n`*** `por` ***`p`***", onde ***`i`*** e ***`j`*** representam, respectivamente, o identificador da ***`i`***-ésima e da ***`j`***-ésima ordem. Logo em seguida, caso existam ordens concluídas, a saída deve ter o formato "`ordem` ***`k`*** `concluida`", onde ***`k`*** é o identificador da

ordem. Caso as duas ordens sejam concluídas após uma mesma transação, elas devem ser impressas ordenadamente de acordo com o identificador k.

Exemplo

No exemplo abaixo, os retângulos internos correspondem à saída que seu programa deve gerar. Observe que seu programa deve seguir a política do **preço igual** durante a leitura do caso de teste.

```
V 2 850.00
V 5 850.00
C 2 780.00
C 1 850.00
```

```
ordem 1 vende para ordem 4 a quantidade 1 por 850.00
ordem 4 concluida
```

```
V 1 800.00
C 3 850.00
```

```
ordem 1 vende para ordem 6 a quantidade 1 por 850.00
ordem 1 concluida
ordem 2 vende para ordem 6 a quantidade 2 por 850.00
ordem 6 concluida
```

```
V 2 780.00
```

```
ordem 7 vende para ordem 3 a quantidade 2 por 780.00
ordem 3 concluida
ordem 7 concluida
```

```
#
```

Sugestões

É de se esperar que vários investidores lancem no sistema ordens de compra com preços baixos. Pode se esperar também o cenário inverso, onde vários investidores lançam no sistema ordens de venda com preços altos. Isso é devido à volatilidade (i.e., variação do preço) das criptomoedas.

Por motivos de simplicidade, sugere-se que sejam criados dois vetores **BUY** e **SELL**, representando, respectivamente, a tabela de negociações de compra e a tabela de negociações de venda.

Observações

- Neste laboratório é obrigatório o uso de **vetores dinâmicos ordenados**, ou seja, os vetores devem dobrar de tamanho quando estão cheios e reduzir o seu tamanho pela metade quando estiverem com apenas um quarto em uso. Além disso, **não é permitido o uso da função `realloc`**.
- Para as turmas A, B e C, esse laboratório tem peso 1.
- Será necessário implementar os arquivos:
    - **`lab02.c`** - deve conter a função principal
    - **`vetor.h`** - estrutura e protótipos de funções
    - **`vetor.c`** - implementação das funções
    - **`ordem.h`** (opcional) - estrutura que define os atributos de uma ordem
- Observações sobre SuSy:
    - Versão do GCC: 4.4.7 20120313 (Red Hat 4.4.7-17)
    - *Flags* de compilação: `-g -ansi -pedantic-errors -Wall -Werror -lm`
    - Utilize comentários do tipo `/* comentário */`;
      comentários do tipo `//` serão tratados como erros pelo SuSy
    - **Tempo máximo de execução: 2 segundo**
- Além das observações acima, esse laboratório será avaliado pelos critérios:
    - Indentação de código
    - Organização
    - Corretude