→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

Trabalho 1 - Simulador de Carteira de Ações

Atenção: atualizado 06/08/2021 (informações sobre README)

Fonte das imagens: Wikipedia

Arquivos disponibilizados:
https://drive.google.com/file/d/1REqtXGdmblMyBtM5tc88oLUdG6MGDonh/view?usp=sharing

O objetivo deste trabalho será praticarmos ordenação, pesquisa e processamento de dados em geral.

A especificação ficou grande (devido aos detalhes), mas a ideia geral é simples.

Ações

Obs: muito do que está sendo informado aqui foi bastante simplificado (e parte das informações pode não ser precisa). As informações aqui passadas não são recomendações de compra nem venda de qualquer ação.

Ações representam basicamente “pedaços” de empresas. Ao comprar, por exemplo, uma ação do Banco do Brasil o investidor passa a ter um pequeno pedaço do banco. As ações são identificadas pelo “ticker”, uma string com 4 letras e um número (normalmente 3, 4 e 11). Por exemplo, o ticker do Banco do Brasil é BBAS3, a Alupar é negociada por meio da ALUP3, ALUP4 e ALUP11 (neste trabalho vamos considerar que os 3 tickers são empresas diferentes), etc.

O investidor pode ser recompensado (ou não) de duas formas ao possuir uma ação: proventos (dividendos e Juros Sobre Capital Próprio -- JCP) e valorização das ações.

Proventos: parte do lucro da empresa pode ser distribuído aos acionistas por meio de proventos (assim, se o BB decidir distribuir R$0.50 de dividendos e uma pessoa possui 1000 ações do BB → receberá 500 reais como participação no lucro). JCPs funcionam de forma

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

parecida e, para simplificar, neste trabalho consideramos que todos proventos funcionam da mesma forma (vamos supor que tudo é dividendo).

Valorização das cotas: quando a empresa lucra (ou quando há expectativa de aumentos futuros de lucros) há uma tendência das ações se valorizarem. Exemplo: se você é dono de metade de uma padaria e sua empresa estiver lucrando cada vez mais → pode ser que o valor de suas ações cresça com o tempo. Essa valorização pode ocorrer mesmo se sua padaria não estiver distribuindo lucro para os sócios (ela pode estar reinvestindo o lucro no próprio negócio (comprando novos equipamentos, abrindo filiais, etc) e, portanto, pode ser que no futuro ela fique cada vez mais lucrativa e eventualmente distribua uma grande quantidade de dividendos para os acionistas).

Com o tempo, por alguns motivos as ações também podem ser agrupadas (grupamentos) ou separadas (splits). Por exemplo, quando ocorre um split 11:10 a cada 10 ações que uma pessoa possui ela receberá uma extra (assim, uma pessoa com 150 ações passará a ter 150+15 = 165 ações). Quando ocorre um split 1:10 (nesse caso chamado de grupamento) → cada 10 ações virará 1 (uma pessoa com 150 ações passará a ter 15). Normalmente essas operações não dão lucro ou prejuízo para o acionista: se uma pessoa possui 2 ações (que possui 4 ações no total) e há um split 2:1 → a pessoa terá 4 ações de uma empresa que agora possui 8 ações no total (ou seja, ela continuará dona de 50% da empresa) e, assim a tendência será que cada uma dessas 4 ações passem a valer metade do que valiam antes.

Neste trabalho, você deverá simular uma carteira de ações. Dado um banco de dados com o histórico de cotações diárias de várias ações (na verdade os preços variam ao longo do dia -- mas para simplificar vamos considerar apenas o preço de fechamento de cada dia), o histórico de eventos (distribuição de dividendos e grupamentos/splits) e um histórico de operações (compra e venda de ações), você deverá simular a evolução dessa carteira mês a mês. Exceto por algumas pequenas simplificações, tudo seguirá o que ocorre na realidade (seu trabalho poderia ser facilmente modificado, por exemplo, para apurar os lucros de operações, patrimônio e dividendos recebidos com o objetivo de declarar um imposto de renda).

Histórico de cotações

Um dos argumentos do seu programa será um arquivo de texto CSV (Comma Separated Values) contendo o histórico (real) de preço de várias ações todos os dias (exceto finais de semana e feriados, quando a Bovespa não funciona) de um período de tempo. Arquivos CSV são formas simples de representar tabelas (cada linha é uma linha de uma tabela e as colunas são separadas por vírgula) e podem inclusive ser salvos e abertos usando planilhas eletrônicas (como o Excel, Google Spreadsheet, LibreOffice Calc, etc).

A seguir temos um exemplo contendo 4 linhas de um arquivo de cotações:

```
ITSA4,2010-01-04,6.557911,6.816703,6.557911,6.811197,3.926552,9604587
ITSA4,2010-01-05,6.849741,6.860753,6.723098,6.789172,3.913855,9147468
ITSA4,2010-01-06,6.767147,6.822210,6.695567,6.706579,3.866241,9212667
```

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

ITSA4,2010-01-07,6.690060,6.712085,6.612973,6.668035,3.844020,7699833

O primeiro valor de cada linha representa o ticker da empresa. A seguir temos a data da cotação. A coluna 6 indica o preço de fechamento da ação no dia (Exemplo: no dia 04/01/2010 a ITSA4 (Itaú S.A.) fechou cotada a R$6.81. (os outros valores representam mínima do dia, máxima, etc -- eles podem ser ignorados no trabalho).

Assim, se uma pessoa comprar ou vender uma ação da ITSA4 nesse dia → o valor da compra/venda será R$6.81 por ação. As linhas do arquivo poderão estar em qualquer ordem (empresas misturadas, datas embaralhadas, etc).

Você deverá fazer o parsing desse arquivo e armazenar os dados (apenas os relevantes) dele no seu programa para consulta (tais consultas deverão ser feitas de forma eficiente).

Para evitar problemas de arredondamento (e erros de precisão de ponto flutuante), seu programa deverá armazenar os preços das ações em centavos (usando **inteiros**): leia os valores como double, multiplique por 100 e trunque para um inteiro. Assim, ITSA4 seria cotada a 681 centavos no dia 04/01/2010.

Antes de truncar os números para armazená-los em seu programa some um pequeno valor real a eles (ex: 0.000001). Isso deverá ser feito para evitar erros devido à multiplicação de ponto flutuante. Por exemplo, talvez o número 5.12 ao ser multiplicado por 100 se transforme em 511.99999999999 (devido à representação de floats). Ao truncar esse valor viraria 511. (somar 0.000001 reduz a chance desse tipo de erro -- mas não resolve sempre).

Obs: os arquivos CSV foram baixados do Yahoo Finance (exemplo: https://br.financas.yahoo.com/quote/ITSA4.SA/history?period1=1262304000&period2=1627603200&interval=div%7Csplit&filter=div&frequency=1d&includeAdjustedClose=true) . A única diferença entre os CSVs disponibilizados nesse site e os do trabalho é que arquivos de várias empresas foram agrupados em um só (e foi adicionada uma coluna com o ticker da empresa) e as linhas foram embaralhadas. Assim, seu trabalho será testado tanto com dados reais do mercado de ações do Brasil quanto com alguns criados especificamente para testes.

Histórico de splits/grupamentos

Um outro arquivo a ser processado pelo seu programa contém o histórico de splits de ações.

Ele também é um arquivo CSV (os dados também foram extraídos do Yahoo Finance) e contém o histórico de splits e grupamentos (assim como o arquivo anterior, os dados não estão ordenados).

Abaixo temos um exemplo de parte desse arquivo:

ABEV3,2013-11-12,5:1
ABEV3,2010-12-20,5:1

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

B3SA3,2021-05-06,3:1
BBDC3,2013-03-26,11:10
VULC3,2016-03-17,1:4
B3SA3,2021-05-17,3:1

Nesse exemplo, quem começou o dia 12/11/2013 com 10 ações da ABEV3 passou a ter 50 ações (split 5:1). Assuma que os splits/grupamentos (e também dividendos) são feitos antes da bolsa de valores começar as negociações do dia (normalmente isso ocorre às 10 da manhã). Ou seja, se uma pessoa tinha 10 ações da ABEV3 no dia 11/11/2013 e comprou 3 ações no dia 12/11/2013 → as 10 ações serão transformadas em 50 ações no dia 12/11/2013 pela manhã e a pessoa compraria 3 ações durante o dia 12/11, ou seja, terminaria o dia 12/11 com 53 ações.

No caso da VULC3, ela sofreu um grupamento 1:4 , ou seja, quem começou o dia 17/03/2016 com 40 ações passou a ter 10.

Caso a divisão de um grupamento não dê um número inteiro (o mesmo vale para um split -- exemplo, um split 11:10), assuma que as frações das ações são perdidas (ou seja, quem tinha 9 ações da VULC3 passaria a ter 2 ações). Na prática, quando isso ocorre as empresas normalmente vendem a parte fracionada das ações e passam o dinheiro ao acionista (por simplicidade, você não deverá simular isso -- na prática isso faria pouca diferença nos resultados em uma carteira com uma quantidade razoável de ações).

Histórico de proventos (dividendos e jcp)

Seu programa também receberá como argumento um arquivo CSV com o histórico de pagamento de proventos das empresas. Esse arquivo também não estará ordenado.

Exemplo de conteúdo:

ITSA4,2016-12-01,0.013547
IRBR3,2020-04-17,0.20975
IRBR3,2020-08-17,0.10162
ITSA4,2010-01-04,0.016381

Nesse exemplo, quem tinha 1 ação ITSA4 no dia 04/01/2010 recebeu 0.016381 real de dividendos. Assuma que os dividendos são pagos logo pela manhã (antes das ações sofrerem splits ou das pessoas comprarem mais ações durante o dia). Assim, uma pessoa que tinha 1000 ações ITSA4 no dia 04/01/2010 antes da bolsa abrir receberá R$16.381 de dividendos independentemente se ITSA4 for sofrer split nesse dia (virando, digamos, 5000 ações) ou se a pessoa comprar mais ações no final desse dia.

No seu programa você deverá fazer os cálculos dos dividendos recebidos e armazená-los em centavos inteiros. Porém, como algumas empresas pagam valores próximos a 1 centavo os cálculos deverão ser feitos (considerando os dividendos pagos e o número de ações do investidor) com double **antes** de truncá-los para inteiros (lembrando de somar um pequeno número real antes de truncar). Exemplo: no caso de 1000 ações ITSA4 acima, você deveria

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

armazenar no seu programa 1638 centavos de dividendos pagos (não 1000 centavos, que seria o valor obtido se tivesse transformado R$0.016381 em centavos inteiros antes de multiplicar pelo número de ações).

Lista de operações de compra/venda

Por fim, uma outra entrada do seu programa será um arquivo CSV contendo na primeira linha um código especificando o formato da saída (isso será explicado depois). A seguir, haverá uma lista de operações.

Há três tipos de operações suportadas pelo seu programa: operações de compra (representadas pelo caractere C) de ações, de venda (representadas por V) e consultas (representados por Q). As consultas serão explicadas posteriormente.

Exemplo de entrada (nesse caso, ele está ordenado por data -- mas não há garantias que o arquivo de entrada estará ordenado). O caractere D é o código especificando o formato da saída:

```
D
2021-07-19,C,ITSA4,100
2021-07-21,C,ITSA4,100
2021-07-22,V,ITSA4,100
2021-07-22,C,PETR4,250
```

Nesse exemplo, no dia 19/07/2021 foram compradas 100 ações ITSA4 (o preço de compra será a cotação para esse dia (em centavos) que foi lida no arquivo de cotações). Em geral, o preço das ações varia ao longo do dia (no seu programa vamos supor que a pessoa sempre paga/recebe o preço de fechamento). No dia 21/07 foram compradas mais 100 e no dia 22/07 foram vendidas 100 ações. Além disso, nesse mesmo dia foram compradas 250 ações PETR4.

Por simplicidade, assuma que NÃO há operações de compra e venda DA MESMA AÇÃO no mesmo dia (isso seria considerado *Day Trade*, e seu programa não precisa tratar).

Todas operações realizadas serão válidas (exemplo: uma pessoa não tentará vender mais ações do que possui, não tentará vender/comprar/consultar em um dia no qual não temos cotação, etc).

Cálculo do custo de compra

O custo de compra é calculado com base no valor pago pelas ações e no estoque de ações que o investidor tem em carteira. Ele é, obviamente, calculado de forma individual para cada empresa que o investidor possui.

Esse valor é utilizado para calcular o lucro obtido com vendas de ações. Além disso, ele também é utilizado para informar o patrimônio do investidor na declaração de imposto de renda.

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

Considere as seguintes operações como exemplo de cálculo (a cotação (fictícia), ou seja, o preço de fechamento de cada ação, em centavos, está entre parênteses).

2021-07-19,C,ITSA4,20 (preço: 10)
2021-07-21,C,ITSA4,10 (preço: 13)
2021-07-22,V,ITSA4,10 (preço: 15)
2021-07-23,C,ITSA4,10 (preço: 9)

- No dia 19/07 foram compradas 20 ações por 10 centavos cada. Logo, ao final do dia 19/07 a pessoa terá 20 ações em carteira com um custo de compra de 200 centavos.
- No dia 21/07 foram compradas 10 ações por 13 centavos cada. No final desse dia a pessoa terá mais 10 ações por um custo de 130 centavos, totalizando 30 ações por 330 centavos (preço médio de compra de cada ação seria 330/30 = 11 centavos).
- No dia 22/07 foram vendidas 10 ações por 15 centavos cada (totalizando 150 centavos de venda). Como o custo médio das 10 ações foi 110 centavos, a pessoa teve um lucro de 150-110 = 40 centavos com a operação e ficou com 20 ações restantes em carteira (com um custo de compra de 330-110=220 centavos). Observe que o preço de venda não altera o custo de compra das ações.
- No dia 23/07 foram compradas 10 ações por 9 centavos. Assim, nesse dia a pessoa passará a ter 20+10 = 30 ações custando 220 + 90 = 310 centavos.

Saídas a serem geradas

O arquivo de operações começa com um caractere indicando o formato da saída a ser gerada pelo seu programa. Esse caractere poderá ser D (saída Detalhada), Q (apenas consultas), F (apenas valores totais ao final da simulação) ou M (valores totais mês a mês). Na mesma linha, poderá ter um caractere adicional R indicando que os dividendos deverão ser reinvestidos.

Arquivos que começam com Q possuem apenas operações desse tipo. Esse será o formato mais fácil de ser implementado (assim, sugiro que comecem a implementação por essa parte). A implementação desse tipo de consulta será útil para as outras etapas do trabalho.

Formato Q:

Caso o arquivo comece com Q, todas as linhas seguintes **terão apenas operações desse tipo**. Você deverá simplesmente imprimir a cotação das ações (em centavos, inteiros) nos dias consultados (sua saída deverá estar na mesma ordem das consultas). Neste trabalho assuma que as consultas são feitas apenas para dias nos quais temos dados.

Exemplo de entrada e saída:

Entrada (arquivo com operações)	Saída esperada (com base no histórico de cotações):
Q	636

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

2010-01-19,Q,ABEV3 2010-01-29,Q,ABEV3 2010-01-19,Q,B3SA3 2010-01-22,Q,ABEV3	599 1350 635

Arquivos que começam com D, F ou M:
Nesse caso, o arquivo de operações conterá apenas operações de compra e venda de ações (não terá consultas). Seu programa deverá simular a evolução de uma carteira de ações (vazia até o momento da primeira compra).

Caso o arquivo de operações comece com D, seu programa deverá gerar um relatório bem detalhado da simulação (deverá seguir exatamente o modelo provido como exemplo). Arquivos começando com F ou M apresentam menos detalhes (mas a simulação completa deverá ser feita da mesma forma!).

Seu programa deverá imprimir no último dia (no qual a bolsa estava aberta) de cada mês a data desse último dia útil e duas tabelas (todas com cabeçalho -- siga os modelos). A impressão deve começar no mês no qual houve a primeira compra de ação da carteira. (ou seja, se a primeira compra ocorreu em 2015-04-02 → a simulação deverá começar nesse dia) e terminar no último mês para o qual temos cotações.

A primeira tabela representará as informações do mês atual. Ela terá em cada linha os dados de cada uma das ações que a pessoa já teve em sua carteira (mesmo se ela tiver comprado e vendido todas ações).

Cada linha terá o ticker, o número de ações dessa empresa em carteira atualmente, o valor atual dessas ações (ou seja, o valor no fechamento do último dia útil do mês), o preço total de custo dessas ações (calculado conforme explicado acima), o total de dividendos recebidos no mês devido a essa ação (note que uma mesma empresa pode pagar mais de uma vez no mesmo mês -- queremos apenas o total), e o lucro obtido devido a operações de venda dessas ações (nesse mês).

Sua tabela deverá estar ordenada em ordem decrescente de lucro total obtido por cada ação no mês, ou seja, lucro de dividendos e lucro com vendas (caso haja empate, desempate pelo ticker em ordem lexicográfica crescente, ou seja, ABEV3 deverá vir antes de BBAS3). Note que, como os cálculos são feitos em centavos (mas no final o resultado é impresso em reais), pode ser que BBAS3 apareça antes de ABEV3 mesmo aparentemente tendo o mesmo total de lucro na tabela exibida (caso tenha lucrado mais, considerando os centavos). Ao final, deve-se ter uma linha informando o total de lucros com dividendos e operações.

A segunda tabela representa os totais desde o início da simulação para cada ação da carteira. Ela contém, para cada ação, o ticker, quantidade, valor atual (patrimônio), custo de compra (esses primeiros valores são iguais aos da outra tabela), valorização (ou seja, a diferença entre

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

o valor atual e o custo de compra), lucros com operações até o momento, dividendos já pagos pela ação e total de rentabilidade (operações + dividendos + valorização). Ela deverá estar em ordem decrescente de rentabilidade total (e ticker, em ordem crescente, no caso de empate).

Após essa tabela deverá haver uma linha com os totais das colunas (exceto das duas primeiras, que não fazem sentido ter totais).

Na pasta exemplos\consultasSimples\consultasDFM há um exemplo bem simples desse tipo de consulta (como as ações foram compradas no mês 05/2021 e temos preços das cotações apenas até 07/2021 → a simulação cobre apenas 3 meses). Nesse exemplo os dados de entrada estão ordenados (para facilitar o entendimento), mas não há garantias disso nos testes do submitty. Veja esse exemplo para entender a simulação e ver como sua saída deverá ser formatada.

Explicação sobre a saída (para facilitar, vamos considerar os valores em reais -- mas seu programa deve fazer o cálculo em centavos):

- A primeira operação ocorre no mês 05/2021 e, portanto, a simulação vai começar nesse mês.
- Mês 05:
    - No dia 2021-05-12 compramos 100 ENBR3. Como a cotação desse dia é 18.27 → o custo de aquisição será R$ 1827.00.
    - Nesse mês, ENBR3 pagou R$1.12345 de dividendos nos dias 12 e 13 (pela manhã). Como as 100 ações foram compradas no dia 12 (sempre no final do dia) → a pessoa receberá apenas o segundo dividendo, totalizando 112 reais (note que no seu programa deverá ser armazenado 11234 centavos (inteiros!) de dividendos -- os 112 reais aparecem apenas ao imprimir).
    - No final do mês ENBR3 estava custando R$ 18.65, logo, o valor das 100 ações será 1865 reais. Assim, a valorização será de 38 reais até o final deste mês.
- Mês 06:
    - No dia 04/06 são compradas 10 ações BBAS3 a um preço de R$35.75. Logo, o custo de aquisição será R$357.50.
    - No dia 04/06 há dividendos e split do BBAS3, mas como isso ocorre pela manhã (antes da compra) → as ações compradas não serão influenciadas por isso (se a pessoa já tivesse ações do BBAS3 antes de comprar mais, a quantidade que ela tinha antes seria influenciada pelos splits e dividendos).
    - No dia 05/06 BBAS3 paga 2 reais de dividendo → Logo, há um crédito de 20 reais de dividendos para as 10 ações.
    - No dia 06 BBAS sofre split 10:1 e, assim, a carteira passará a ter 100 ações BBAS3 (o custo total de aquisição não é alterado).
    - No dia 07 BBAS paga 3 reais de dividendo → Há um crédito de 300 reais em dividendos para as 100 ações que a carteira tem agora.
    - No dia 07 ENBR paga 1 real de dividendo → Há um crédito de 100 reais de dividendos pelas 100 ações.

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

    - No dia 07 BBAS sofre um novo split (splits ocorrem pela manhã após os dividendos) 10:1 → Agora a carteira possuirá 1000 ações BBAS3.
    - No dia 08 ENBR paga 1 real de dividendo pela manhã → Crédito de 100 reais de dividendos.
    - No dia 08, 50 ações ENBR3 são vendidas (à tarde) por R$18.80 (cada) → agora há 50 ações em carteira. O valor de venda foi 940 reais. O preço médio de compra das 100 ações que havia em carteira era R$1827.00, ou 913.50 para as 50 ações. Logo, há um lucro de R$940 - 913.50 = R$26.50 com a operação de venda. Como saíram 50 ações com um preço de COMPRA de R$913.50, as 50 ações que restam em carteira terão custo de COMPRA de 1827.00 - 913.50 = 913.50.
    - No dia 09, 100 ações ENBR3 são compradas por R$18.70. Agora, teremos 50+100 = 150 ações com preço de compra total de 913.50 + 1870.00 = R$ 2783.50.
    - Assim, no final do mês teremos:
        - 1000 BBAS3 com preço de compra de R$ 357.50
        - 320 reais de dividendos vindos de BBAS3
        - 150 ENBR3 com preço de compra R$2783.50
        - 200 reais de dividendos vindos de ENBR3.
        - 26.50 ganhos de lucro com vendas (na verdade, apenas uma venda) de ENBR3.
- Mes 07:
    - Nesse mês não há eventos de compra, venda, splits ou dividendos.
    - Sendo assim, os únicos valores que mudam são o “Valor Atual” de cada ação (que varia devido à cotação delas nos meses em questão). Como o valor das ações muda, o patrimônio total da pessoa também muda.

Se alguma ação fosse completamente vendida, ela continuaria aparecendo na tabela (mas quantidade, custo e valor atual seriam 0 -- operações e dividendos da segunda tabela podem não ser 0).

No caso de relatórios do tipo M, seu programa deve fazer a simulação normalmente, mas imprimir apenas o lucro total obtido em cada mês (com dividendos e operações). No caso de relatórios do tipo F, seu programa também deve fazer a mesma simulação, mas imprimir apenas os totais do final da simulação (patrimonio final, custo de compra final, soma de dividendos recebidos, etc), ou seja, apenas os dados que aparecem na última linha gerada pelo relatório do tipo D (mas com outra legenda). Veja os exemplos.

Em “exemplos\dadosReais” há exemplos com cotações/splits/dividendos reais. “exemploMGLU3.csv” simula um investimento de 10000 reais em MGLU3 em 14/12/2015 (Isso teria se transformado em mais de 7 milhões de reais!)

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

Se uma pessoa tivesse comprado 10,000 reais em MGLU3 em 14/12/2015, vendido em 2020-11-06 e imediatamente trocado por TAEE4 (uma boa pagadora de dividendos), até o final de 2021-07 ela teria recebido R$1,329,661 de dividendos de TAEE4 (exemploMGLU3_TAEE11.csv) e ficado com um patrimônio de 11 milhões de reais (sem contar os dividendos recebidos). Claro, é muito fácil fazer essas simulações aqui (difícil é prever isso…)

Arquivos contendo R após o D, F ou M:
Nesse caso, a única diferença é que os dividendos recebidos em cada dia deverão ser imediatamente (no final do dia) reinvestidos nas próprias empresas que os pagaram. Assim, por exemplo, se em um dia sua carteira receber 500 reais de dividendos do BBAS3 e cada ação custa 30 reais, você deverá comprar no final desse dia 16 ações BBAS3 (assuma que os 20 reais restantes sejam gastos pelo investidor). Com isso, teremos o efeito de "juros compostos".

Os dividendos reinvestidos devem ser tratados como uma compra normal. Ou seja, no exemplo acima é como se o investidor tivesse recebido 500 reais de dividendos e no final do dia tivesse feito uma compra normal de 16 ações BBAS por 30 reais cada (o resultado será o mesmo que seria obtido se tivesse uma linha "DATA,C,BBAS3,16" no arquivo de operações, onde DATA é a data onde o dividendo foi pago).

Em "exemplos\consultasSimples\consultasDFM" há um exemplo onde há reinvestimento de dividendos (operacoesDR.csv). Esse é o mesmo exemplo passado na seção anterior -- a única diferença é que há reinvestimento.

Por exemplo, no dia 13/05 ENBR3 paga 11234 de dividendos para o investidor. Como a cotação da ação nesse dia é 1840 centavos, ele comprará 6 ações ENBR3 no final do dia (note que a pessoa pode também fazer outras compras nesse mesmo dia -- a compra com os recursos de dividendos é adicional). Assim, ele passará a ter 100+6= 106 ações a um preço de compra de R$1827.00 + R$110.40 = R$1937.40.

Se uma empresa pagar dividendos várias vezes durante o mesmo dia, o valor desses dividendos deverá ser somado e usado para uma única compra no final do dia. Assim, se BBAS paga 10 reais + 10 reais e fechar o dia custando 15 reais → será comprada 1 ação no final do dia (não 0+0).

Em "exemplos/dadosReais" há dois arquivos (exemploOperacoesD.txt e exemploOperacoesDR.txt) comparando o efeito de se reinvestir dividendos. No caso da SAPR4, foram investidos 10 mil reais em 2010. Em 2021, se os dividendos não tivessem sido reinvestidos a pessoa teria tido um lucro total de 74 mil reais. Se tivessem sido reinvestidos, o lucro final seria 105 mil reais.

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

Como seu programa será executado

O caminho para os arquivos de, respectivamente, preços, splits, dividendos e operações será dado como argumentos (nessa ordem) de linha de comando para seu programa.

Exemplo de execução:

```
./a.exe exemplos/consultasSimples/consultasDFM/precos.csv exemplos/consultasSimples/consultasDFM/splits.csv
exemplos/consultasSimples/consultasDFM/dividendos.csv
exemplos/consultasSimples/consultasDFM/operacoesF.csv
```

A saída deverá ser impressa em stdout (saída padrão) usando o *cout*.

Eficiência do seu programa, dados de entrada, etc

Para facilitar, assuma que a carteira de ações terá no máximo 100 ações diferentes. Assuma que os arquivos de preços, splits, proventos e operações terão, no máximo, 500 mil linhas.

Seu programa deverá processar tais dados de forma eficiente e não poderá ter nada com complexidade quadrática em função do tamanho (número de linhas) dos arquivos de entrada. Ou seja, se houver P linhas no arquivo de preços e O linhas no arquivo de operações, complexidades do tipo O(P^2), O(PO), O(O^2) deixarão seu código excessivamente ineficiente (além dos pontos dos testes, seu trabalho perderá MUITOS pontos na avaliação manual).

O que poderá ser utilizado

A princípio, você poderá utilizar apenas as bibliotecas string, iostream, cassert (recomendo que use asserts para detectar erros), fstream, iomanip (para formatar as tabelas) e sstream. Se achar que outra biblioteca é realmente necessária, pergunte ao professor se o uso será permitido.

Vectors, algoritmos prontos de ordenação, maps (e similares) não deverão ser usados, ou seja, tudo deverá ser implementado pelo aluno. A ideia será praticar alocação dinâmica de memória (para os arrays), uso de strings, pesquisa e ordenação.

Dicas

- Use *getline* e um *stringstream* para facilitar o parsing dos arquivos de entrada.
- Para facilitar, use strings (C++) em vez de arrays de caracteres (estilo C).
- Como já sabemos o tamanho máximo dos arquivos de entrada, o aluno poderá declarar arrays (com alocação dinâmica, para evitar stack overflow) com esse tamanho máximo a fim de facilitar a implementação.
- Sugiro que faça uma cópia deste documento e marque de vermelho/sublinhado as partes que achar mais importante (para que possa verificar depois se seu programa está seguindo toda a especificação).
- Sempre faça vários exemplos e diagramas em papel.
- Crie casos de teste simples para validar sua implementação e entender melhor o problema. Isso é uma importante habilidade de computação.

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

- Seu programa fará basicamente a mesma simulação feita pelo site https://www.meusdividendos.com/smartfolio

Organização do código

Seu código deverá ser bem organizado. Sugiro que crie classes para deixar seu código bem organizado. Por exemplo, uma para armazenar as cotações e permitir que o preço de uma determinada ação seja consultado , uma para armazenar detalhes (quanto de dividendos essa empresa já pagou, quantas dessas ações atualmente temos, etc) de cada ação em carteira, etc.

Cada classe deverá estar definida em arquivos separados (a definição em um arquivo .h e a implementação em um .cpp).

Arquivo README

Seu trabalho deverá incluir um arquivo README.

Tal arquivo conterá (nessa ordem):

- Seu nome/matricula
- Informações sobre todas fontes de consulta utilizadas no trabalho.
- Um pequeno parágrafo (no máximo 100 palavras) bem sucinto explicando o que você fez para obter um código eficiente. Exemplo de explicação sucinta (mas sem sentido): "O ponto chave do meu trabalho foi ordenar todos os dados usando o algoritmo da bolha (que tem complexidade O(n)). A seguir, utilizei criei um banco de Dados Relacional indexado para armazenar os preços, dividendos e splits e permitir consultas eficientes. As operações de compra e venda foram processadas …."

Submissao

Submeta seu trabalho utilizando o sistema Submitty até a data limite. Seu programa será avaliado de forma automática (os resultados precisam estar corretos, o programa não pode ter erros de memória, etc), passará por testes automáticos "escondidos" e a qualidade do seu código será avaliada de forma manual.

A avaliação automática será feita em partes. Por exemplo, haverá casos de teste apenas com consultas de cotações, casos de teste pequenos (onde algoritmos ineficientes seriam aprovados), casos apenas com operações de compra e casos mais desafiadores. Assim, um programa que não segue completamente a especificação poderá obter uma nota parcial.

Você deverá todos os arquivos .cpp (e .h) e o README.txt. Seu programa será compilado com o comando "g++ *.cpp -O3".

Duvidas

Dúvidas sobre este trabalho deverão ser postadas no PVANET Moodle. Se esforce para implementá-lo e não hesite em postar suas dúvidas!

→ LEMBREM-SE DE USAR PAPEL E CANETA COMO RASCUNHO ANTES DE IMPLEMENTAR ←

Avaliacao manual

Principais itens que serão avaliados (além dos avaliados nos testes automáticos):
- Comentarios
- Indentacao
- Nomes adequados para variáveis
- Separação do código em funções lógicas
- Uso correto de const/referencia
- Uso de variáveis globais apenas quando absolutamente necessário e justificável (uso de variáveis globais, em geral, e' uma má prática de programação).
- Organização do código em classes
- etc

Regras sobre plágio e trabalho em equipe

Leia as regras gerais aqui:
https://docs.google.com/document/d/1qwuZtdioZO-QiDsq6SAm7m-DU6bLrid7_7nEL4g9HOk/edit?usp=sharing