

MC202 — ESTRUTURAS DE DADOS

Laboratório 04 — Tipo Abstrato de Dados

Problema

Você acaba de ser contratado por uma das corretoras mais importantes do Brasil. Como qualquer grande empresa, a carteira de clientes que eles administram é muito grande e, por isso, é necessário administrar automaticamente os investimentos feitos para cada cliente. Como novo programador da equipe, você deve criar um programa que faça um relatório das ações contidas no portfólio de um cliente. Para integrar os seus sistemas, a empresa usa tipos de dados abstratos. Assim, você receberá as interfaces `acao.h` e `portfolio.h` e deve implementar os arquivos correspondentes, `acao.c`, `portfolio.c` e `lab04.c`.

Entrada

A primeira linha contém o nome e o sobrenome do cliente, seguidos do número **N** ($1 \leq N \leq 100$) de ações que ele possui em seu portfólio. Cada uma das **N** linhas seguintes representa um investimento feito em uma ação. Cada linha contém o nome da ação; em seguida, a quantia em reais inicialmente investida; seguida de um inteiro **M** ($1 \leq M \leq 20$) que representa quantas vezes o valor dessa ação sofreu variação; e, para finalizar, uma lista de **M** números reais que representam cada uma das variações percentuais relatadas, em que um número positivo representa um ganho e um número negativo representa uma perda.

Exemplo:

```
Hari Seldon 3
Petrobras 1000.00 4 3.0 -5.3 -6.2 -2.1
Itau 600.50 2 5.0 9.2
Bradesco 750 3 5.0 -5.0 0.25
```

Saída

A primeira linha da saída contém uma linha da forma "`Relatorio de nome sobrenome`". As **N** linhas seguintes resumem as perdas e ganhos de cada uma das ações, na mesma ordem de entrada. Cada linha é composta pelo nome da ação, a quantia remanescente na ação usando duas casas decimais e uma string "`GANHO`" no caso em que a quantia atual for maior que o valor investido, e "`PERDA`" caso contrário.

Exemplo:

```
Relatorio de Hari Seldon
Petrobras 895.72 PERDA
Itau 688.53 GANHO
Bradesco 750.00 PERDA
```

Observações gerais

No SuSy, haverá 3 tipos de tarefas com siglas diferentes para cada laboratório de programação. Todas possuirão os mesmos casos de teste. As siglas são:

1. **SANDBOX:** Esta tarefa serve para testar o programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém arquivos submetidos aqui **não serão corrigidos**.
2. **ENTREGA:** Esta tarefa tem limite de **uma única submissão** e serve para entregar a versão final dentro do prazo estabelecido para o laboratório. Não use essa tarefa para testar o seu programa: submeta aqui quando não for mais fazer alterações no seu programa.
3. **FORAPRAZO:** Esta tarefa tem limite de **uma única submissão** e serve para entregar a versão final após o prazo estabelecido para o laboratório, mas com nota reduzida (conforme a ementa). O envio nesta tarefa irá substituir a nota obtida na tarefa ENTREGA apenas se o aluno tiver realizado as correções sugeridas no feedback ou caso não tenha enviado anteriormente em ENTREGA.

Observações sobre SuSy:

- Versão do GCC: C-ANSI 4.8.2 20140120 (Red Hat 4.8.2-15).
- Flags de compilação:
`-ansi -Wall -pedantic-errors -Werror -g -lm`
- Utilize comentários do tipo `/* comentário */;`
comentários do tipo `//` serão tratados como erros pelo SuSy.

Além das observações acima, esse laboratório será avaliado pelos critérios gerais:

- Indentação de código e outras boas práticas, tais como:
 - uso de comentários (apenas quando forem relevantes);
 - código simples e fácil de entender;
 - sem duplicidade (partes que fazem a mesma coisa).
- Organização do código:

- tipos de dados criados pelo usuário e funções bem definidas e tão independentes quanto possível.
- Corretude do programa:
 - programa correto e implementado conforme solicitado no enunciado;
 - inicialização de variáveis sempre que for necessário;
 - dentre outros critérios.