

Considere uma empresa X de vendas cujos funcionários (vendedores) possuem cada qual sua matrícula identificadora de venda, ou seja, cada vez que um deles efetua uma venda é registrado no lote da data da venda uma nova linha contendo consecutivamente o valor de sua matrícula e o valor da venda efetuada.

Em anexo a este arquivo existe uma pasta intitulada `2013/` que contém doze subpastas referentes aos grupos de lotes mensais gerados pela empresa X no ano passado. A quantidade de lotes de cada mês é variante assim como o total de entradas (vendas) de cada lote.

O arquivo `funcionarios.dat` (também em anexo) armazena em cada linha uma matrícula identificadora de funcionário seguida do respectivo nome.

Construa um *script* em Haskell que processe a pasta `2013/` e produza um relatório de vendas da empresa X (arquivo de saída intitulado `report.dat`) contendo uma tabela em quatro colunas,

1. Matrícula identificadora do funcionário
2. Nome do funcionário
3. Total de vendas que este funcionário efetuou
4. Valor total em vendas deste funcionário

A tabela em `report.dat` deve aparecer ordenada ascendentemente pelo valor total arrecadado por funcionário (quarta coluna).

ROTEIRO DE IMPLEMENTAÇÃO

Construa,

1. (1.0) Defina uma estrutura `Report` que represente um relatório de vendas.
2. (2.0) Construa função `procLote` que processa o conteúdo de um lote de entrada (`String`) e introduza seus resultados (por fusão) num `Report` de entrada.
3. (2.0) Construa função `procListLote` que processa uma lista de conteúdos de lotes (`[String]`) e introduza seus resultados num `Report` de entrada.
4. (2.0) Construa função `reportToStr` que converta um `Report` de entrada numa string com quebras de linha (relatório de vendas tabelado).
5. (5.0) Construa o aglutinador de ações `main` o qual deve receber o nome da pasta a ser processada, via linha de comando (repasse `2013/`), e gerar `report.dat` com o relatório de vendas. Nesta etapa deve-se processar os lotes em todas as pastas, gerar um `[String]` dos conteúdos selecionados, submetê-lo a `procListLote`, depois o resultado desta à `reportToStr` e, por fim, gravar o conteúdo retornada por esta última no arquivo `report.dat`.