

# Prova

## Algoritmos e Estruturas de Dados I

Prof.: Carlos Camarão

13 de Dezembro de 2005

Em todas as questões, você pode supor a existência de métodos estáticos de entrada e saída de valores de tipos básicos, definidos em uma classe `EntradaPadrão`. Por exemplo, para ler inteiros, você pode usar `EntradaPadrão.lêInteiro()`. Esse método lê um inteiro do dispositivo de entrada padrão e retorna o valor lido, de tipo `int`.

1. (6 pontos) Escreva um programa que leia um valor inteiro positivo  $n$ , calcule o valor dado por  $\sum_{i=1}^n (-i+1)/(3*i)$ , e imprima esse valor. Decomponha o seu programa em três partes, cada uma delas implementada por um método: leitura, cálculo do valor desejado a partir do valor lido, e impressão do valor calculado.

O programa deve tratar a exceção que ocorre quando a cadeia de caracteres lida não representa um número inteiro, informando o usuário com uma mensagem apropriada e realizando a operação de leitura novamente.

A operação de divisão não pode ser uma “divisão inteira” (não pode ter como parâmetros dois números inteiros), mas sim uma divisão de números de ponto flutuante (i.e. deve ser realizada depois de converter pelo menos um dos números inteiros para um número de ponto flutuante).

2. (7 pontos) Uma operação *anulsub* aplicada a uma lista de  $n$  números inteiros positivos  $x = [a_0, a_1, \dots, a_i, \dots, a_{n-1}]$  — onde o elemento  $a_i$  tem índice  $i$  na lista, para  $i$  de 0 a  $n-1$  — e a um inteiro positivo  $i$  (chamado de *índice* da operação), onde supõe-se  $0 \leq i < n-1$ , retorna como resultado a lista  $[a_0, \dots, a_{i-1}, 0, a_i + a_{i+1}, a_{i+2}, \dots, a_{n-1}]$ .

Aplicando  $n-1$  operações *anulsub* a uma lista de  $n$  inteiros positivos quaisquer obtém-se uma outra lista, chamada de *lista alvo*. Por exemplo, se *anulsub*( $x, i$ ) representa a aplicação da operação a uma lista  $x$  e um índice  $i$ , temos:

$$\begin{aligned} \text{anulsub}([12, 10, 4, 3, 5], 1) &= [12, 0, 14, 3, 5] \\ \text{anulsub}([12, 0, 14, 3, 5], 2) &= [12, 0, 0, 17, 8] \\ \text{anulsub}([12, 0, 0, 17, 8], 1) &= [12, 0, 0, 17, 8] \\ \text{anulsub}([12, 0, 0, 17, 8], 0) &= [0, 0, 0, 17, 8] \end{aligned}$$

Faça um programa que:

- leia um valor inteiro positivo  $n$  (o número de inteiros da lista original),  $n$  valores inteiros positivos da lista  $a_0, \dots, a_{n-1}$  e, em seguida, uma seqüência de  $n-1$  valores que indicam os índices de operações *anulsub*,
- realize operações *anulsub* consecutivamente como no exemplo acima, e
- imprima a lista alvo.

Se não existir lista alvo (i.e. se a entrada estiver de alguma forma incorreta), o programa deve emitir uma mensagem de erro apropriada.

**3.** (7 pontos) Um armazém trabalha com  $n$  mercadorias diferentes ( $n = 100$ ), identificadas por números de 1 a  $n$ .

Um funcionário desse armazém tem salário mensal estipulado como 20% da receita mensal obtida com as vendas que ele realizou, mais um bônus igual a 10% da receita mensal obtida com o produto que teve o maior número de unidades vendidas.

Escreva um programa que leia, para cada mercadoria de 1 a  $n$ , o número de unidades vendidas em um determinado mês por esse funcionário e o preço unitário da mercadoria, e imprima o salário do funcionário, nesse mês.

Repetindo: para leitura dos dados, você pode supor que existam métodos estáticos *readInt* e *readFloat*, que não têm parâmetros e retornam respectivamente um valor do tipo `int` e um valor do tipo `float`, sendo o valor lido do dispositivo de entrada padrão.