

1ª Prova de Programação de Computadores

Professor: Carlos Camarão

14 de Outubro de 2004

1. (7 pontos) Escreva um programa que leia um valor inteiro positivo n e imprima em uma linha os valores de cada um dos n primeiros números perfeitos, separados um do outro por um ou mais espaços. Um número perfeito é um número inteiro positivo que é igual à soma dos seus divisores. Exemplo: $6=1+2+3$ e $28=1+2+4+7+14$ são números perfeitos.

Dica: você pode ou imprimir diretamente os números perfeitos encontrados ou concatená-los em uma cadeia de caracteres (i.e. em um “string”) e depois imprimir a cadeia de caracteres (não esquecendo de separar um número do seguinte por um ou mais espaços).

2. (6 pontos) Escreva um programa para contagem de votos em uma eleição presidencial, da qual participam dois candidatos. Os dados de entrada do programa representam os votos dos eleitores, codificados como a seguir. Um valor 1 representa um voto para o primeiro candidato, e um valor 2 para o segundo candidato; 3 representa um voto nulo e 4 representa um voto em branco.

O seu programa deve ler os dados de entrada, até que um valor 0 (zero) seja lido, e calcular e imprimir o total de votos para cada candidato, a porcentagem de votos nulos e a porcentagem de votos em branco.

Suponha que os dados são lidos do dispositivo de entrada padrão. Você pode supor que existe um método `EntradaPadrão.lêInteiro()` que lê e retorna um valor inteiro lido desse dispositivo. Mais precisamente, você pode supor que existe uma classe `EntradaPadrão` na qual é definido um método com “assinatura” `public static int lêInteiro()`. Esse método pode ser chamado simplesmente usando `EntradaPadrão.lêInteiro()`.

3. (7 pontos) Escreva um programa que leia repetidamente valores inteiros positivos n e h , um após o outro, o primeiro representando o número de um aluno, e o segundo representando a sua altura em centímetros, até que um valor inteiro não positivo (para o número ou para a altura) seja lido. O programa deve determinar qual é o aluno mais alto e o mais baixo, imprimindo o número do aluno mais alto e sua altura, e o número do aluno mais baixo e sua altura.

Dica: Use `Integer.MAX_VALUE` como valor inicial da menor altura e número do aluno com menor altura.