

Prova de Programação de Computadores

Prof.: Carlos Camarão

23 de Junho de 2005

1. (10 pontos) Escreva um programa que leia um seqüência de números inteiros positivos do dispositivo de entrada padrão e imprima, no dispositivo de saída padrão, essa seqüência em ordem inversa. A seqüência de números lidos termina quando um número negativo ou igual a zero for lido. Esse número negativo ou zero não faz parte da seqüência de números positivos da entrada, sendo usado apenas para indicar o final da seqüência de números positivos, e não deve ser impresso. Por exemplo, se a entrada for 1 2 3 0, a saída deve ser 3 2 1.

Para ler os valores inteiros da entrada, você deve usar o método `nextInt` da classe `Scanner`, e deve desconsiderar mas tratar dados incorretos, que não representam inteiros válidos. Por exemplo, se a entrada for 1 2 * 3 -1, a saída deve ser 3 2 1.

Para isso, você deve tratar a exceção `InputMismatchException`, causada pelo método `nextInt` quando o dado que foi lido não constitui um inteiro, mas continuar a leitura, desconsiderando o dado que causou a exceção.

2. (10 pontos) Escreva um programa em Java que leia uma seqüência de cadeias de caracteres (valores de tipo `String` do dispositivo de entrada padrão e imprima, no dispositivo de saída padrão, uma seqüência de caracteres 0 ou 1, sendo o i -ésimo caractere dessa seqüência igual a 1 se a i -ésima seqüência lida é palíndromo, e 0 caso contrário.

Uma seqüência é palíndromo se ela é igual ao seu inverso. Por exemplo, `k`, `xax` e `*1aba1*` são palíndromos.

A leitura deve terminar quando o método `hasNext` da classe `Scanner` retornar valor `false`.

Dica: Para resolver esse problema, você pode usar, se quiser, o método `reverse` da classe `StringBuilder`, que retorna a seqüência de caracteres passada como argumento em ordem inversa.

A classe `StringBuilder` é semelhante a `String`, representa também uma seqüência de caracteres, com a diferença de que os caracteres da seqüência podem ser modificados (e inseridos e removidos), ao contrário das seqüências de caracteres representadas por objetos da classe `String`. Nessa questão não está

mos interessados diretamente nessa propriedade de modificação (nem de inserção nem de remoção) de caracteres, mas sim no fato de que a classe *StringBuilder* define um método *reverse*, sem parâmetros, que você pode usar. O método *reverse* dessa classe retorna um objeto de tipo *StringBuilder* que tem os caracteres da sequência representada pelo objeto alvo da mensagem (trata-se de um método não estático) em ordem inversa. Por exemplo, *new StringBuilder("123").reverse()* retorna um objeto de tipo *StringBuilder* que representa a mesma sequência de caracteres que *new StringBuilder("321")*.

Para comparar igualdade de dois objetos de tipo *StringBuilder*, você tem (infelizmente) que transformá-los em objetos de tipo *String*, usando o método *toString* da classe *StringBuilder*; em seguida, você pode comparar os dois valores de tipo *String* obtidos, usando o método *equals* da classe *String*.

3. (10 pontos) Escreva em Java um programa que leia dois valores inteiros positivos *numQuest* e *numAlunos*, que representam respectivamente o número de questões de uma prova de múltipla escolha e o número de alunos de uma turma.

Em seguida, o programa deve ler, para cada uma das questões, um valor inteiro que indica a resposta correta de cada questão.

Depois, o programa deve ler, para cada um dos *numAlunos* alunos, *numQuest* valores inteiros não-negativos, que representam as respostas de cada aluno para cada questão da prova.

O programa deve imprimir, para cada aluno, o seu número (de 1 a *numAlunos*) e o número de respostas corretas dadas por esse aluno.

Além disso, o programa deve imprimir a média das notas da turma nessa prova.

Você pode considerar que os dados de entrada estão corretos.