Prova de Programação de Computadores

Prof.: Carlos Camarão

3 de Dezembro de 2008

1. Escreva um programa que leia, do dispositivo de entrada padrão, um número inteiro positivo n, em seguida n números inteiros $v_1, \ldots, v_n$, e então crie e inicialize um arranjo v de tamanho n de modo que v`[`i`-1]=`v_i (ou seja, cada posição i de v deve conter um dos dos n inteiros lidos v_i), para $i = 1, \ldots, n$.

   Em seguida o programa deve ler números inteiros k e, se k não estiver entre 1 e n, a execução do programa deve terminar, caso contrário v_k deve ser impresso (v_k é o k-ésimo inteiro lido após n, ou seja, v`[`k`-1]`).

2. Escreva um programa que leia inteiros contidos em seqüência de linhas $l_1, \ldots, l_n$ — onde cada linha l_i ($i = 1, \ldots, n$) contém um número inteiro positivo k_i e em seguida k_i números inteiros, sendo cada inteiro na linha separado do seguinte por um ou mais espaços — e imprima uma seqüência de linhas l'_i, para $i = 1, \ldots, n$, onde cada linha l'_i contém k_i e os k_i números inteiros de l_i mas em ordem inversa.

   O programa deve prever a possibilidade de ocorrência de dados incorretos. Dados incorretos podem ocorrer pelo fato de uma linha conter uma seqüência de caracteres que não representa um inteiro, ou que o número de inteiros de uma linha l_i não ser igual ao valor do primeiro inteiro k_i que ocorre na linha. No caso de ocorrência de dado incorreto na linha l_i , uma linha com a seqüência de caracteres:

   ```
   "*** Erro na linha i ***"
   ```

   deve ser impressa — em vez da linha l_i com os inteiros na ordem inversa (onde i representa o número da linha).

   A existência de uma seqüência de caracteres que não representa um inteiro deve ser tratada por um *tratador de exceção*, causada por chamada ao método *nextInt* definido na classe *Scanner*. Uma chamada a *nextInt* propaga a exceção *InputMismatchException* caso o valor lido não constitua um valor inteiro válido.

   Para testar o fim dos dados de entrada, você pode:

   - usar chamada ao método *hasNext* da classe *Scanner*, que retorna `false` se e somente se não há mais dados a serem lidos, ou
   - tratar a exceção *NoSuchElementException*, propagada por chamada a *nextInt* quando não há mais dados de entrada.

3. Uma chamada a *Math.random*`()` retorna um valor (pseudo-)aleatório de tipo `double` entre 0 e 1.

   Um número (pseudo-)aleatório entre 1 e um valor inteiro n maior que 1 pode ser obtido usando:

   Math.floor`(`*Math.random*`() * (n+1))`

   Defina uma função *numAleatEntre1E* que, ao receber como argumento um valor inteiro n maior que 1, retorna um número (pseudo-)aleatório entre 1 e n.

   No programa descrito a seguir chamadas a *numAleatEntre1E* devem ser feitas para representar o resultado obtido ao jogar para cima um *dado* comum com faces numeradas de 1 a 6.

   Escreva um programa que leia um número inteiro positivo n e imprima quantas vezes cada face do dado foi obtida como resultado do ato de jogar um dado para cima n vezes — o ato de jogar o dado para cima corresponde a (deve ser simulado como) uma chamada a *numAleatEntre1E*`(6)`.