**Ex. 1** Uma sequência é dita *monotônica crescente* se, para qualquer par de elementos da sequência, $e_{i+1} \geqslant e_i$. Posto de outra forma, a partir do primeiro elemento, os elementos da sequência crescem, ou permanecem iguais aos anteriores, mas nunca decrescem. Para uma sequência *monotônica decrescente* temos que $\forall\, i \bullet e_{i+1} \leqslant e_i$ . Uma sequência não monotônica é dita *aleatória*.

Escreva um programa que recebe uma sequência de números naturais. A sequência é terminada por `-1`. Ao final da sequência seu programa indica que a sequência é monotônica crescente (imprime na tela "Crescente"), monotônica decrescente (imprime na tela "Decrescente"), ou se a sequência não é monotônica (imprime na tela "Aleatória").

Escreva duas funções, a função `mais()` recebe dois números naturais como argumentos e retorna `TRUE` se o primeiro argumento é maior ou igual ao segundo, e retorna `FALSE` do contrário. A outra função, `menos()` recebe dois números naturais como argumentos e retorna `TRUE` se o primeiro argumento é menor ou igual ao segundo, e retorna `FALSE` do contrário.

O programa principal lê a sequência de números, chama as duas funções a cada novo elemento da sequência, e ao final, escreve na tela qual o tipo da sequência. Sequências vazias e unitárias são aleatórias. *Pista*: quais operadores lógicos são associativos?

Exemplo de execução:

```
Digite a sequência:
2 3 5 5 5 6 7 -1
Crescente
```

Outro exemplo de execução:

```
Digite a sequência:
2 3 5 4 5 6 7 -1
Aleatória
```

**Ex. 2** Uma *redução* é uma operação que "reduz" um conjunto de números para um único valor. Somatórios, produtórios e médias são exemplos de reduções.

Escreva um programa que efetua reduções por soma (somatórios) ou por produto (produtórios) de conjuntos de números. Seu programa principal lê do teclado um código de operação (**1** para soma, **2** para produto) e um conjunto de números naturais. O conjunto é terminado por $-\mathbf{1}$.

Escreva duas funções, a função `soma()` recebe o valor acumulado da soma até o momento e mais um valor e acrescenta este valor ao acumulado; a função `prod()` recebe o produto parcial e multiplica o novo valor pelo produto parcial. Ao final do conjunto, seu programa principal imprime a operação (soma ou produto) e o resultado da redução. O programa então espera para computar a próxima redução.

Exemplo de execução:

```
Escolha a operação:
2
Digite a sequência:
2 2 2 2 -1
O produto eh:  16
```

**Vire a folha.**

**Ex. 3** Escreva uma função que decide se um número inteiro é primo. Se o número for primo, sua função deve retornar TRUE, do contrário deve retornar FALSE.

**Ex. 4** Escreva um programa que imprime a lista de todos os números primos entre 1 e 50.