

# 5 Estruturas de Repetição

1. A biblioteca *math.h* oferece algumas funções matemáticas. Supondo a inexistência desta biblioteca, implemente uma função para calcular:
  - a) o piso de um número real, tal como a função *floor*;
  - b) o teto de um número real, tal como a função *ceil*;

Em seguida, chame estes módulos a partir de um módulo *main* para testar seu programa.

2. Implemente uma função para calcular a potência entre dois números inteiros quaisquer. Em seguida, chame este módulo a partir de um módulo *main* para testar seu programa. Teste as potências de  $2^3$ ,  $2^{-3}$ ,  $-3^2$  e  $-3^{-2}$ .
3. Um número primo é um inteiro positivo que possui apenas dois divisores também positivos diferentes. Escreva uma função para retornar verdadeiro ou falso para o fato de um número ser primo. Em seguida, chame este módulo a partir de um módulo *main* para testar seu programa.
4. Implemente uma função para calcular o fatorial de um número. O fatorial de 0 é 1, por definição. Para qualquer outro número inteiro positivo  $n$ , o seu fatorial é dado por  $n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1$ . Em seguida, chame este módulo a partir de um módulo *main* para testar seu programa.
5. A sequência infinita 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55... é conhecida como a sequência de Fibonacci. Nela, os dois primeiros números são 0 e 1. Os demais números são obtidos pela soma dos dois números imediatamente anteriores. Implemente uma função retornar o  $n$ -ésimo termo da sequência de

Fibonacci. Em seguida, chame este módulo a partir de um módulo *main* para testar seu programa.

6. Finja que os únicos operadores disponíveis na linguagem C sejam o incremento ( $++$ ) e o decremento ( $--$ ). Implemente funções para calcular:
- a) o oposto;
  - b) o valor absoluto;
  - c) a subtração;
  - d) a adição;
  - e) a multiplicação;
  - f) a divisão;
  - g) o resto da divisão.

Estas funções devem apenas considerar valores inteiros. Em seguida, chame este módulo a partir de um módulo *main* para testar seu programa.

7. Um cálculo aproximado para a raiz quadrada de um número positivo  $n$ , segue o seguinte algoritmo:
- i) Estabelecer um intervalo de atuação entre  $a=0$  e  $b=n$
  - ii) Calcular o ponto médio  $c$  do intervalo
  - iii Se  $c^2 > n$ , reduzir o intervalo de atuação fazendo  $b = c$ ; caso contrário, reduzir o intervalo de atuação fazendo  $a = c$
  - iv) Repetir os passos ii) e iii) até que o tamanho do intervalo de atuação torne-se muito pequeno, por exemplo, menor que  $10^{-6}$

Escreva uma função para estimar uma aproximação para a raiz quadrada de um número positivo. Em seguida, chame este módulo a partir de um módulo *main* para testar seu programa.

8. Complemente a resolução de cada exercício anterior desta seção, escrevendo soluções com recursividade.