

# Programação Procedimental

Maria Adriana Vidigal de Lima

Universidade Federal de Uberlândia - 2013

## 1 Escrita de Algoritmos

### 1.1 Técnicas de Programação e Algoritmos

Na elaboração de programas maiores e complexos é necessário utilizar um método sistemático de programação que permita a obtenção de programas corretos, confiáveis, flexíveis e eficientes. Os passos a seguir estabelecem uma metodologia de programação:

- a) Análise do problema.
- b) Projeto do programa: algoritmo e estruturas de dados.
- c) Codificação em linguagem de programação e teste do programa.
- d) Manutenção e atualização do programa.

Um **algoritmo** é uma sequência finita de instruções básicas, definidas sem ambiguidade e executáveis em tempo finito. A execução desta sequência resolve um problema computacional, definido a partir de uma estratégia definida na análise do problema.

Um algoritmo pode ser descrito através de uma linguagem algorítmica (pseudocódigo) ou fluxograma. No fluxograma, cada operação básica é representada por um desenho, e em linguagem algorítmica cada operação básica é escrita em linguagem próxima à linguagem natural, considerando-se algumas regras comuns às linguagens de programação. Um algoritmo pode ser implementado em qualquer linguagem de programação e essa implementação pode ser trivial, como uma simples transcrição de operações básicas, ou complexa, dependendo das características da linguagem de programação escolhida e dos tipos de dados nela definidos.

Do ponto de vista lógico, um algoritmo é constituído por três estruturas lógicas:

- **sequencial**: quando diversos comandos devem ser executados numa sequência linear.
- **seletiva**: quando a ação a ser executada depende de uma inspeção ou teste.
- **repetitiva**: quando um conjunto de ações deve ser executado repetidamente enquanto uma determinada condição permanecer válida.

A descrição de um algoritmo se baseia em estruturas lógicas que organizam as instruções que o compõem. Além disso, deve ser especificada a sua função, quais são os dados de entrada e/ou saída. Por exemplo, o Algoritmo 1 descreve o problema de calcular a soma entre dois números.

Todos os algoritmos são descritos por funções, sendo que a função básica (denominada Principal) é sempre a primeira a ser executada. A função Principal é seguida de parênteses, que delimitam uma lista de argumentos, que são informações que a função necessita para que possa executar a tarefa associada. Neste caso a função Principal não possui argumentos de entrada.

### 1.2 Comandos em Linguagem Algorítmica

Uma sequência de comandos válidos da linguagem forma um bloco de instruções. Os comandos válidos devem estar separados pelo símbolo ";".

---

**Algoritmo 1** - Adição de dois números

---

Objetivo: Dados dois números, calcular e retornar a sua soma.

Variáveis: A, B, RESULTADO: inteiro

**Principal** ()

**Início**

**leia**(A, B);

  RESULTADO  $\leftarrow$  A + B;

**imprima**(RESULTADO);

**Fim**

---

### 1.2.1 Comando de atribuição

Uma atribuição é indicada pelo operador  $\leftarrow$ . A expressão do lado direito do operador é calculada e o resultado é armazenado na variável ao lado esquerdo. Por exemplo:

```
x  $\leftarrow$  50/y;  
z  $\leftarrow$  p * 3.2 + q;
```

### 1.2.2 Leitura dos valores de entrada do algoritmo:

O comando **leia** deve ser utilizado para que uma variável possa ter um valor associado, a partir de uma leitura do teclado. O mesmo comando pode ser utilizado para uma lista de variáveis, neste caso os valores são lidos um de cada vez e atribuídos às variáveis de acordo com a posição das mesmas na lista. Exemplos:

```
leia nota1, nota2;  
leia x, y, z;
```

### 1.2.3 Impressão dos resultados do algoritmo:

O comando **imprima** deve ser utilizado para a impressão de mensagens ou de valores calculados ou obtidos pelo algoritmo. A impressão dos resultados pode também ser mesclada com uma mensagem explicativa. Os valores são impressos, um a um, na ordem em que aparecem na lista. Exemplos:

```
imprima "bom dia";  
imprima x1, x2, x3;  
imprima a + b/c;  
imprima "O resultado e ", m + n;
```

### 1.2.4 Comandos de seleção:

Os comandos de seleção permitem selecionar quais comandos serão executados dependendo de uma condição. Os comandos de seleção uni e bidirecional têm o seguinte formato:

```
se <condição>  
  então <bloco>  
fim se
```

```
se <condição>  
  então <bloco1>  
  senão <bloco2>  
fim se
```

O comando **se-então** avalia a condição, que pode resultar em *Verdadeiro* ou *Falso*. Se a condição é verdadeira, o bloco de comandos será executado, senão este bloco é descartado. O comando **se-então-senão** define dois blocos de instruções, um para cada caso.

---

**Algoritmo 2** - Maior entre dois números

---

Objetivo: Dados dois números, retornar o maior.

Variáveis: A, B, max: inteiro

**Principal** ()

**Início**

  leia (A, B);

  se  $A > B$ ;

    então  $max \leftarrow A$ ;

    senão  $max \leftarrow B$ ;

  fim se

  imprima "maior = ", max

**Fim**

---

### 1.2.5 Comandos de repetição:

Os comandos de repetição podem definir iterações controladas por variáveis ou com controle lógico.

O comando **repita para** utiliza variável de controle e é representado por:

```
repita para i de 1 até n
  <bloco>
fim repita
```

em que bloco é uma sequência de uma ou mais instruções. No exemplo a seguir, o bloco de instruções entre os comandos **repita para** e **fim repita** é executado repetidamente de acordo com a variável controladora.

```
repita para  $i \leftarrow 1$  até 10
```

```
  imprima  $i, i * i$ ;
```

```
fim repita
```

O comando **repita** pode ser utilizado para a impressão de uma sequência de números, como no algoritmo:

---

**Algoritmo 3** - Imprimir sequência de 10 números

---

Objetivo: Dado um número, imprimir a sequência dos 10 próximos números.

Variáveis: a,i: inteiro

**Principal** ()

**Início**

  leia (a);

  repita para  $i \leftarrow 1$  até 10

    imprima a;

$a \leftarrow a + 1$ ;

  fim repita

**Fim**

---

O comando **repita enquanto** utiliza um teste lógico e é representado por:

```
repita enquanto <condição> faça
  <bloco>
fim repita
```

O bloco de instruções entre os comandos **repita enquanto** e **fim repita** é executado enquanto a condição definida por uma expressão lógica for satisfeita. Exemplo:

```

i ← 1
repita enquanto i ≤ 10
  imprima i, i * i;
  i ← i + i;
fim repita

```

O comando **repita até** executa pelo menos uma vez o bloco a ser repetido, uma vez que a condição é testada apenas após a sua execução. A expressão a seguir representa o comando **repita até**:

```

repita
  <bloco>
até que <condição>
fim repita

```

A condição a ser testada deve estar associada a variáveis que sejam atualizadas no interior do bloco. Caso isso não ocorra, o programa ficará repetindo indefinidamente o bloco de comandos. Exemplo:

```

leia a;
repita
  t ← a * a;
  imprima a, t;
  leia a;
até que a ≤ 0

```

### 1.3 Exemplos de Algoritmos

1. **Algoritmo de Euclides:** Dados dois números positivos m e n encontre seu maior divisor comum, isto é o maior inteiro positivo que divide tanto m como n. Assuma que m é sempre maior que n, e n diferente de zero.

```

principal ()
início
  leia m, n;
  r = m % n; // resto da divisão de m por n
  repita enquanto r != 0
    m = n;
    n = r;
    r = m % n;
  fim repita
  imprima n;
fim

```

2. Multiplicação de dois números inteiros positivos.

```

principal ()
início
  leia m, n;
  r = 0;
  repita enquanto n != 0
    r = r + m;
    n = n-1;
  fim repita
  imprima r;
fim

```

## 1.4 Exercícios

1. Escreva um algoritmo que leia três números e imprima o maior deles.
2. Escreva um algoritmo que leia três números e os imprima em ordem crescente.
3. Escreva um algoritmo que leia uma certa quantidade de números e imprima o maior deles e quantas vezes o maior número foi lido. A quantidade de números a serem lidos deve ser fornecida pelo usuário. Assuma que o usuário sempre fornecerá um número positivo.
4. Escreva um algoritmo que leia um número inteiro entre 100 e 999 e imprima na saída cada um dos algarismos que compõem o número.
5. Uma empresa paga RS30.00 por hora normal trabalhada e RS45.00 por hora extra. Escreva um programa que leia o total de horas normais e o total de horas extras trabalhadas por um empregado em um ano e calcule o salário médio mensal e o salário anual deste trabalhador.

## Referências

- [1] Salvetti, D. D., Barbosa M. B. *Algoritmos*, Makron Books, 1993.