

## 5 Comando de Seleção Simples e Composta

Ronaldo F. Hashimoto e Carlos H. Morimoto

Essa aula introduz o comando de seleção, que permite ao seu programa tomar decisões sobre o fluxo do processamento, ou seja, dependendo do estado ou condição de seu programa, esse comando permite selecionar que procedimento deve ser executado a seguir.

Ao final dessa aula você deverá saber:

- Utilizar o comando if.
- Utilizar o comando if-else
- Identificar situações onde é necessário utilizar o comando if-else e situações onde é suficiente utilizar o comando if.
- Simular o processamento dos comandos if e if-else em um programa.

### 5.1 Sintaxe

A sintaxe do comando de seleção é ilustrada na figura 3.

A <condição> é uma expressão relacional que tem como resultado um valor **verdadeiro** ou **falso** (veja a aula sobre **fundamentos**). A sequência de comandos dentro do **if** <comando\_1>, <comando\_2>, ..., <comando\_n>, bem como a sequência de comandos dentro do **else** <outro-comando\_1>, <outro-comando\_2>, ..., <outro-comando\_m>, podem ser comandos de atribuição, impressão de mensagens na tela, leitura de números inteiros pelo teclado ou eventualmente um outro comando de seleção.

Seleção Simples	Seleção Composta
<pre>if (&lt;condição&gt;) {   &lt;comando_1&gt;;   &lt;comando_2&gt;;   ...   &lt;comando_n&gt;; }</pre>	<pre>if (&lt;condição&gt;) {   &lt;comando_1&gt;;   &lt;comando_2&gt;;   ...   &lt;comando_n&gt;; }  else {   &lt;outro-comando_1&gt;;   &lt;outro-comando_2&gt;;   ...   &lt;outro-comando_m&gt;; }</pre>

Figura 3: Sintaxe dos comandos de seleção simples e composta.

**Observação:** se dentro do **if** ou do **else** existir apenas um comando, não é necessário colocá-lo entre chaves. A figura 4 mostra a sintaxe do **if** nesse caso.

Seleção Simples	Seleção Composta
<pre>1 if (&lt;condição&gt;) 2 &lt;comando&gt;;</pre>	<pre>1 if (&lt;condição &gt;) 2 &lt;comando&gt;; 3 else 4 &lt;outro_comando&gt;;</pre>

Figura 4: Sintaxe dos comandos de seleção com comando único. Note que as chaves não são necessárias nesse caso.

## 5.2 Descrição

### 5.2.1 Seleção simples

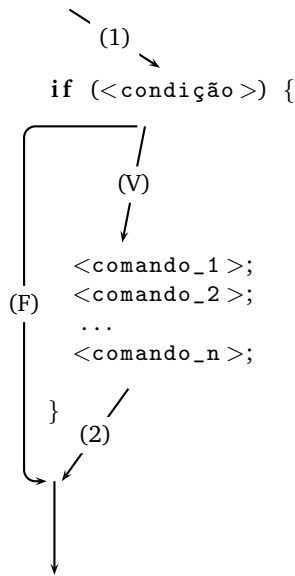


Figura 5: Fluxo do comando de seleção simples.

Basicamente, o comando de seleção simples tem o seguinte significado: se a <condição> for **verdadeira**, a sequência de comandos <comando\_1>, <comando\_2>, ..., <comando\_n> é executada. Caso contrário, a sequência de comandos <comando\_1>, <comando\_2>, ..., <comando\_n> não é executada.

A figura 5 ilustra a execução desse comando em um programa. Primeiramente a <condição> do if é testada (seta marcada com (1)). Se “de cara” a <condição> é **falsa**, o fluxo do programa ignora a sequência de comandos dentro do if e segue a seta marcada com (F). Agora, se a <condição> é **verdadeira**, então o fluxo do programa segue a seta marcada com (V) e executa a sequência de comandos dentro do if; executado o último comando (<comando\_n>), o fluxo do programa segue a seta marcada com (2) e o programa segue adiante.

### 5.2.2 Seleção Composta

O comando de seleção composta tem o seguinte significado: se a <condição> for **verdadeira**, a sequência de comandos <comando\_1>, <comando\_2>, ..., <comando\_n> é executada. Caso contrário a sequência executada é a sequência de comandos <outro\_comando\_1>, <outro\_comando\_2>, ..., <outro\_comando\_m>.

Vamos analisar o “fluxo” do comando de seleção composta if-else através da figura 6. Primeiramente, a execução do programa vem e testa a <condição> do if (seta marcada com (1)). Se a <condição> é **verdadeira**, o fluxo do programa segue a seta marcada com (V) e executa a sequência de comandos dentro do if e ignora a sequência de comandos dentro do else seguindo a seta marcada com (2) a instrução seguinte do comando if-else é executada. Agora, Se a <condição> do if é **falsa**, o fluxo do programa ignora a sequência de comandos dentro do if e segue a seta marcada com (F) e executa a sequência de comandos dentro do else. No final o fluxo segue a seta marcada com (3) executando a instrução seguinte ao comando if-else.

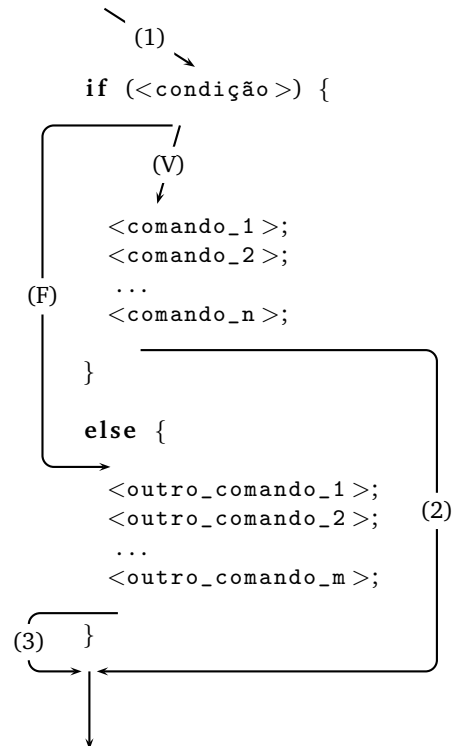


Figura 6: Fluxo do comando de seleção composta.

## 5.3 Exemplos Comentados

### 5.3.1 Exemplo 1

Vamos analisar o “fluxo” de um programa usando o comando de seleção simples if aplicado a um problema bastante simples:

Escreva um programa em C que leia um número inteiro e devolva o seu valor absoluto.

**Solução:** Esse problema apenas ilustra a utilização do comando `if`. Sabemos que para calcular o módulo (valor absoluto) de um número inteiro basta trocar o seu sinal quando ele for negativo. Portanto, uma solução simples para esse problema seria:

1. imprima uma mensagem para o usuário saber o que fazer;
2. leia pelo teclado um número inteiro em `num1`;
3. se `num1` for negativo, transforme-o para positivo;
4. imprima o resultado
5. fim

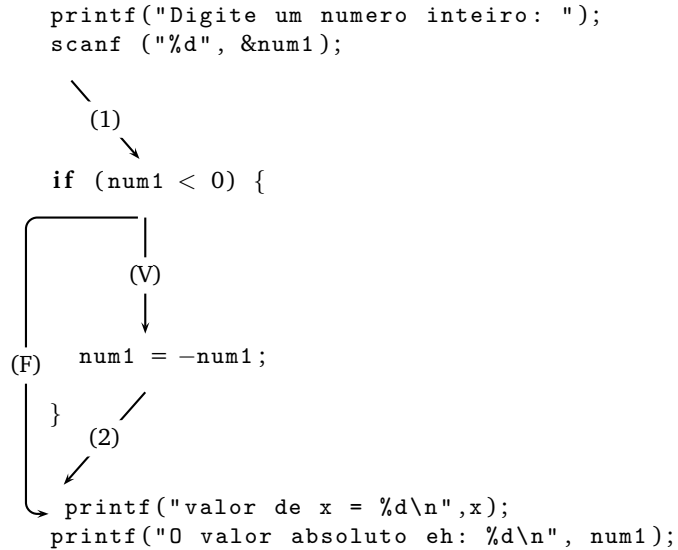


Figura 7: Cálculo do módulo de um número inteiro.

O funcionamento desse programa pode ser entendido pelo diagrama da figura 7. Esse trecho de programa lê do teclado um número inteiro antes do comando `if` e o armazena na variável `num1`. A condição do `if` é então testada e no caso de `num1` ser menor que zero, o programa simplesmente troca o sinal de `num1` e no final imprime o novo valor de `num1`. Observe que se o número digitado for positivo, o sinal de `num1` não é trocado, e seu valor é impresso na tela.

A solução completa é dada abaixo:

```
1 # include <stdio.h>
2 # include <stdlib.h>
3
4 int main () {
5     int num1;
6     printf("Digite um numero inteiro: ");
7     scanf("%d ", &num1);
8
9     if (num1 < 0) {
10        num1 = -num1;
11    }
12
13    printf("O valor absoluto eh: %d\n", num1);
14    return 0;
15 }
```

### 5.3.2 Exemplo 2

Escreva um programa que leia 2 números inteiros e imprima o maior.

#### Solução:

A simplicidade desse programa permite observar o uso do comando if-else. Para isso, vamos utilizar 2 variáveis para armazenar os valores de entrada, e outra para armazenar o maior valor. Uma solução possível pode ser descrita de modo informal como:

1. imprima uma mensagem para o usuário saber o que fazer
2. leia pelo teclado dois números inteiros num1 e num2
3. se num1 maior que num2
  - (a) maior = num1
4. senão
  - (a) maior = num2
5. imprime o conteúdo da variável maior
6. fim

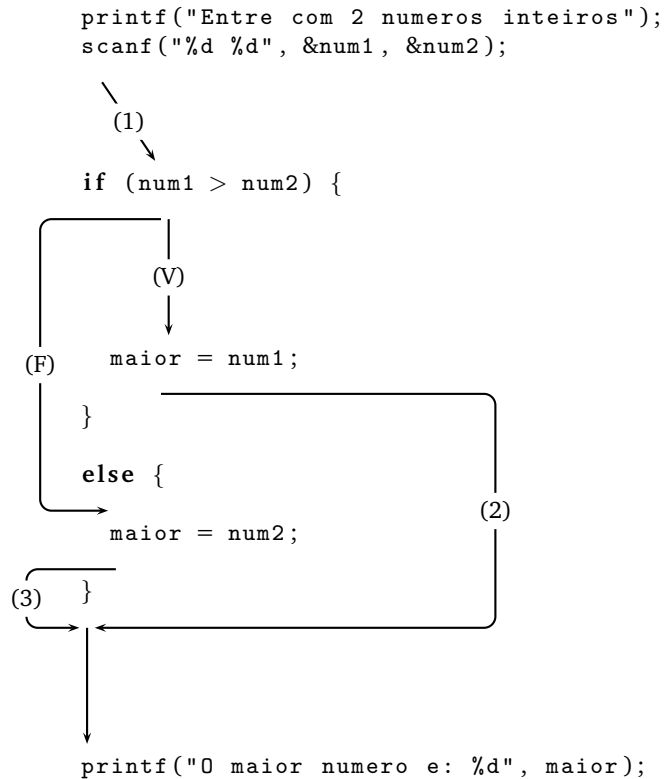


Figura 8: Lê 2 números e imprime o maior.

O funcionamento do programa pode ser entendido também pelo diagrama da figura 8.

A solução completa é dada abaixo:

```
1 # include <stdio.h>
2 # include <stdlib.h>
3
4 int main () {
5     int num1, num2, maior;
6     printf("Entre com 2 numeros inteiros");
7     scanf("%d %d", &num1, &num2);
8
9     if (num1 > num2) {
10        maior = num1;
11    }
12    else {
13        maior = num2;
14    }
15
16    printf("O maior numero e: %d", maior);
17    return 0;
18 }
```

Observe que basta comparar num1 com num2 para sabermos qual o maior.

### 5.3.3 Exemplo 3

Dados um número inteiro  $n > 0$  e uma sequência com  $n$  números inteiros, determinar a soma dos inteiros positivos da sequência. Por exemplo, para  $n=7$  e para a sequência com  $n=7$  números inteiros

6   -2   7   0   -5   84

o seu programa deve escrever o número 25.

#### Solução:

Uma forma possível para resolver esse problema é imaginar uma variável que armazena as somas parciais. Essa variável deve iniciar com o valor zero, e para cada número da sequência, se o número lido é positivo, ser somada com mais esse número, até o final da sequência. Assim, para o exemplo acima, o valor de soma torna-se 6 após processar o primeiro elemento da sequência (soma-se o 6), 13 após o terceiro (soma-se o 7), 21 após o sexto (soma-se o 8), e assim até o final.

Uma solução possível pode ser descrita de modo informal como:

1. imprima uma mensagem para o usuário saber o que fazer
2. leia pelo teclado a quantidade de números da sequência na variável  $n$
3. inicialize uma variável soma com zero
4. enquanto  $n$  maior que zero faça:
  - (a) leia pelo teclado o número da sequência na variável  $num$
  - (b) se  $num$  maior que zero
    - i. acumule na variável soma o número lido
  - (c) diminua 1 da variável  $n$
5. imprima na tela a soma final
6. fim

Uma solução do problema escrito em C:

```
1 # include <stdio.h>
2 # include <stdlib.h>
3
4 int main () {
5     /* declaracoes */
6     int num; /* variavel utilizada para leitura da sequencia */
7     int soma; /* variavel que armazena a soma da sequencia */
8
9     /* programa */
10    printf("Digite a quantidade de numeros da sequencia: ");
11    scanf("%d", &n);
12
13    while (n > 0) {
14
15        printf("Digite um numero: ");
16        scanf("%d", &num);
17
18        if (num > 0)
19            soma = soma + num ;
20
21
```

```
22     }
23
24     printf("Soma dos numeros positivos da sequencia = %d\n", soma);
25     /* fim do programa */
26     return 0;
27 }
```

**Observação:** note que como dentro do `if` temos somente um comando `soma = soma + num`, não é necessário colocá-lo entre chaves.

## 5.4 Exercícios Recomendados

A solução para esses e outros exercícios você encontra na lista de exercícios em <http://www.ime.usp.br/~macmulti/exercicios/inteiros/index.html>.

Procure fazer os exercícios antes de consultar a solução.

1. Dizemos que um número natural é triangular se ele é produto de três números naturais consecutivos.  
Exemplo: 120 é triangular, pois  $4 \cdot 5 \cdot 6 = 120$ .  
Dado um inteiro não-negativo  $n$ , verificar se  $n$  é triangular.