

MAC 2166 – Introdução à Computação

POLI - PRIMEIRO SEMESTRE DE 2007

Material Didático

Prof. Ronaldo Fumio Hashimoto

COMANDO DE SELEÇÃO SIMPLES E COMPOSTA**Sintaxe**

A sintaxe do comando de seleção é:

Seleção Simples	Seleção Composta
<pre>if (<condição>) { <comando_1>; <comando_2>; ... <comando_n>; }</pre>	<pre>if (<condição>) { <comando_1>; <comando_2>; ... <comando_n>; } else { <outro_comando_1>; <outro_comando_2>; ... <outro_comando_m>; }</pre>

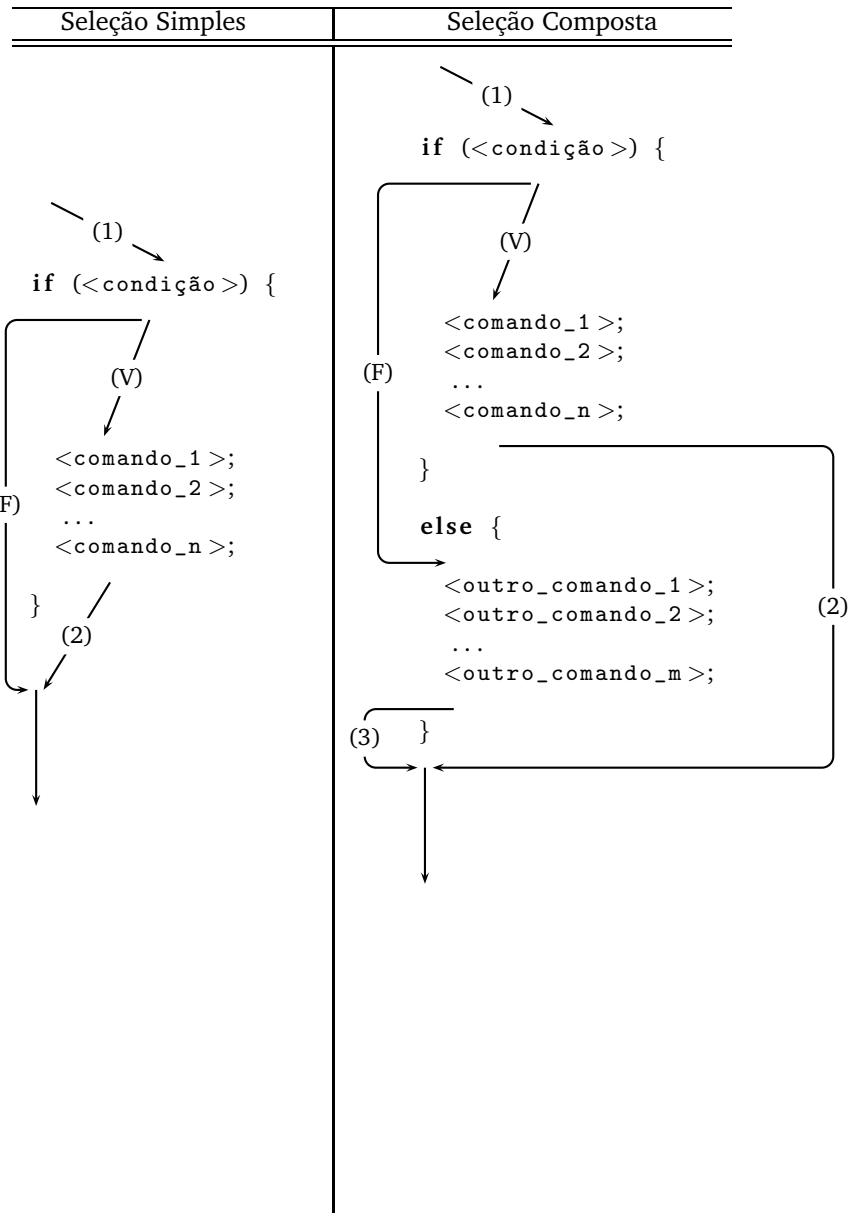
A <condição> é uma expressão relacional que tem como resultado um valor **verdadeiro** ou **falso** (veja aula **comandos básicos**). A seqüência de comandos dentro do **if** <comando_1>, <comando_2>, ..., <comando_n>, bem como a seqüência de comandos dentro do **else** <outro_comando_1>, <outro_comando_2>, ..., <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.

Observação: se dentro do **if** ou do **else** tem somente um comando, não é necessário colocá-lo entre chaves.

Exemplo:

Seleção Simples	Seleção Composta
<pre>if (<condição>) <comando>;</pre>	<pre>if (<condição>) <comando>; else <outro_comando>;</pre>

Descrição



Basicamente, o comando de seleção simples tem o seguinte significado: se a <condição> for **verdadeira**, a seqüência de comandos <comando_1>, <comando_2>, ..., <comando_n> é executada. Caso contrário, a seqüência de comandos <comando_1>, <comando_2>, ..., <comando_n> não é executada.

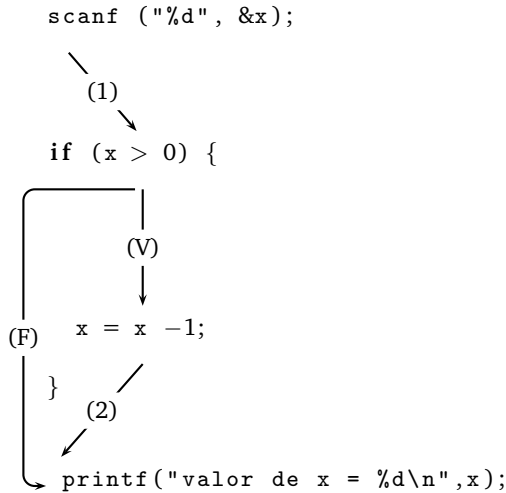
O comando de seleção composta tem o seguinte significado: se a <condição> for **verdadeira**, a seqüência de comandos <comando_1>, <comando_2>, ..., <comando_n> é executada. Caso contrário, a seqüência de comandos <outro_comando_1>, <outro_comando_2>, ..., <outro_comando_m> não é executada.

Fluxo do Comando de Seleção Simples

Vamos analisar o “fluxo” do programa usando o comando de seleção simples if. Primeiramente, a execução do programa vem e testa a <condição> do if (seta marcada com (1)). Se “de cara” a <condição> é **falsa**, o fluxo do programa ignora a seqüência de comandos 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 seqüência de comandos

dentro do `if`; executado o último comando (`<comando_n>`), o fluxo do programa segue a seta marcada com (2) e fluxo do programa segue adiante.

O trecho de programa seguinte lê do teclado um número inteiro e armazena na variável `x`. No caso de `x` se maior que zero, programa simplesmente subtrai 1 de `x` e no final imprime o novo valor de `x`. Caso contrário, ou seja, `x` é um número menor ou igual a zero, o comando de atribuição `x = x - 1`; não é executado e o número `x` é impresso na tela.



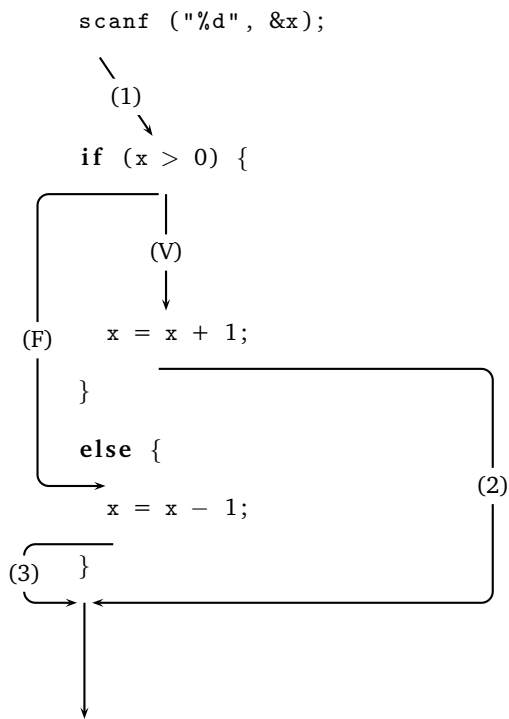
O primeiro comando, a leitura de número inteiro pelo teclado é executado antes do `if` e armazena o número lido na variável `x`. Depois o fluxo do programa segue a seta marcada com (1) e testa a condição (`x>0`) do `if`. Se ela é **verdadeira**, os comandos dentro do `if` são executados (que nesse caso contém apenas a atribuição `x = x - 1`;) seguindo a seta marcada com (V). Depois o fluxo do programa segue a seta marcada com (2). Agora, se a condição (`x>0`) do `if` é **falsa**, o fluxo do programa segue a seta marcada com (F), não executando o comando `x = x - 1`; e a instrução seguinte ao comando `if` é executada.

Fluxo do Comando de Seleção Composta

Analisando o “fluxo” do programa usando o comando de seleção composta `if-else`, vemos que 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 seqüência de comandos dentro do `if` e ignora a seqüência 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 seqüência de comandos dentro do `if` e segue a seta marcada com (F) e executa a seqüência de comandos dentro do `else`. No final o fluxo segue a seta marcada com (3) executando a instrução seguinte ao comando `if-else`.

O trecho de programa seguinte lê do teclado um número inteiro e armazena na variável `x`. No caso de `x` se maior que zero, programa simplesmente soma 1 de `x` e no final imprime o novo valor de `x`. Caso contrário, ou seja, `x` é um número menor ou igual a zero, o comando de atribuição `x = x - 1`; é executado e o número `x` resultante é impresso na tela.



O primeiro comando, a leitura de número inteiro pelo teclado é executado antes do `if` e armazena o número lido na variável `x`. Depois o fluxo do programa segue a seta marcada com (1) e testa a condição $(x > 0)$ do `if`. Se ela é **verdadeira**, os comandos dentro do `if` são executados (que nesse caso contém apenas a atribuição $x = x + 1$;) seguindo a seta marcada com (V). Depois o fluxo do programa segue a seta marcada com (2) para a próxima instrução, ignorando o comando $x = x - 1$; que está dentro do `else`. Agora, se a condição $(x > 0)$ do `if` é **falsa**, o fluxo do programa segue a seta marcada com (F), ignorando o comando $x = x + 1$; e executando o comando $x = x - 1$;. Depois o fluxo segue a seta marcada com (3) executando a próxima instrução que é um comando de impressão na tela.

Exemplos de Comentados

Exemplo 1

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

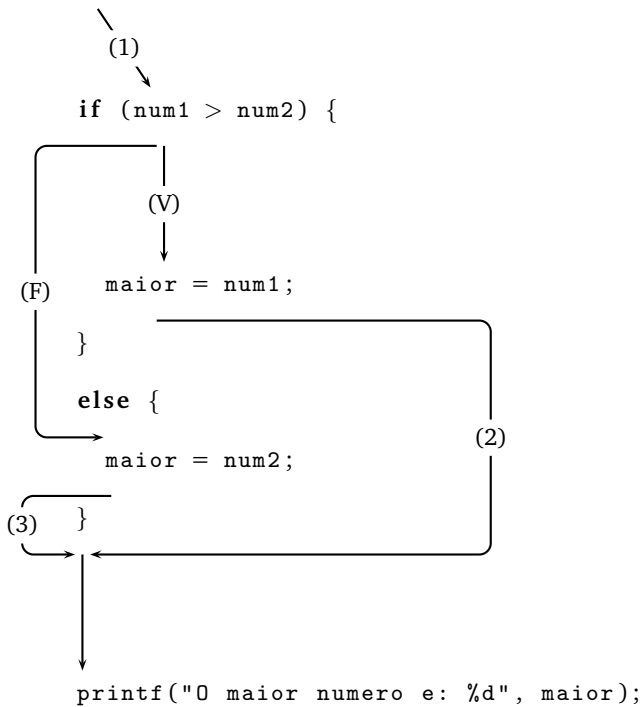
(a) maior = num2

5. imprime o conteúdo da variável maior

6. fim

O funcionamento do programa pode ser entendido também pelo diagrama abaixo:

```
printf("Entre com 2 numeros inteiros");  
scanf("%d %d", &num1, &num2);
```



A solução completa é dada abaixo:

```
# include <stdio.h>  
# include <stdlib.h>  
  
int main () {  
    int num1, num2, maior;  
    printf("Entre com 2 numeros inteiros");  
    scanf("%d %d", &num1, &num2);  
  
    if (num1 > num2) {  
        maior = num1;  
    }  
    else {  
        maior = num2;  
    }  
  
    printf("O maior numero e: %d", maior);  
  
    system("pause");  
    return 0;  
}
```

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

Exemplo 2

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

6, -2, 7, 0, -5, 8, 4

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 seqüência, se o número lido é positivo, ser somada com mais esse número, até o final da seqüência. Assim, para o exemplo acima, o valor de soma torna-se 6 após processar o primeiro elemento da seqüência (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 seqüência na variável n
3. inicialize uma variável soma com zero
4. enquanto n maior que zero faça:
 - (a) leia pelo teclado o número da seqüência 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:

```
# include <stdio.h>
# include <stdlib.h>

int main () {
    /* declaracoes */
    int num; /* variavel utilizada para leitura da sequencia */
    int soma; /* variavel que armazena a soma da sequencia */

    /* programa */
    printf("Digite a quantidade de numeros da sequencia: ");
    scanf("%d", &n);

    while (n > 0) {

        printf("Digite um numero: ");
        scanf("%d", &num);

        if (num > 0)
            soma = soma + num ;

    }

    printf("Soma dos numeros positivos da sequencia = %d\n", soma);
    /* fim do programa */

    system ("pause");
    return 0;
}
```

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

Dúvidas

Dúvidas deste material pode ser enviadas para o “Fórum para assuntos específicos da turma WEB”.