

Comandos de Controle do Programa

- Comandos de Seleção
 - **if, else, switch**
- Comandos de Iteração
 - **for, while, do-while**
- Comandos de Desvio
 - **return, goto, break, continue**

O Comando if

- O **if**
- O **else**
- O **if-else-if**
- A expressão condicional
- **ifs** aninhados
- O operador **?**

O Comando if

- A forma geral de um comando **if** é:

```
if (condição) {  
    seqüência de comandos;  
}
```

- A expressão, na *condição*, será avaliada:
 - Se ela for zero, a declaração não será executada;
 - Se a condição for diferente de zero a declaração será executada.

Exemplo:

```
#include <stdio.h>  
void main ()  
{  
    int num;  
    printf ("Digite um numero: ");  
    scanf ("%d",&num);  
    if (num > 10)  
        printf ("\n\n O numero e maior que 10");  
    if (num == 10)  
    {  
        printf ("\n\n Voce acertou!\n");  
        printf ("O numero e igual a 10.");  
    }  
    if (num < 10)  
        printf ("\n\n O numero e menor que 10");  
}
```

O else

- Podemos pensar no comando **else** como sendo um complemento do comando **if**. O comando **if** completo tem a seguinte forma geral:

```
if (condição) {  
    seqüência_de_comandos_1;  
}  
else {  
    seqüência_de_comandos_2;  
}
```

O else

- A expressão da condição será avaliada:
 - ◆ Se ela for diferente de zero, a *seqüência_comandos_1* será executada.
 - ◆ Se for zero a *seqüência_comandos_2* será executada.
- É importante nunca esquecer que, quando usamos a estrutura **if-else**, estamos garantindo que uma das duas declarações será executada.

Exemplo

```
#include <stdio.h>
void main ( )
{
    int num;
    printf ("Digite um numero: ");
    scanf ("%d", &num);
    if (num == 10)
    {
        printf ("\n\n Voce acertou!\n");
        printf ("O numero e igual a 10.\n");
    }
    else
    {
        printf ("\n\n Voce errou!\n");
        printf ("O numero e diferente de 10.\n");
    }
}
```

O if-else-if

- A estrutura **if-else-if** é apenas uma extensão da estrutura **if-else**. Sua forma geral é:

```
if (condição_1) {
    seqüência_de_comandos_1;
}
else if (condição_2) {
    seqüência_de_comandos_2;
}
...
else if (condição_n) {
    seqüência_de_comandos_n;
}
else {
    seqüência_de_comandos_default;
}
```

O if-else-if

A estrutura acima funciona da seguinte maneira:

- O programa começa a testar as condições começando pela 1 e continua a testar até que ele ache uma expressão cujo resultado dê diferente de zero. Neste caso ele executa a seqüência de comandos correspondente.
- Só uma seqüência de comandos será executada, ou seja, só será executada a seqüência de comandos equivalente à primeira condição que der diferente de zero. A última seqüência de comandos (default) é a que será executada no caso de todas as condições darem zero e é opcional.

Exemplo

```
#include <stdio.h>
void main ()
{
    int num;

    printf ("Digite um numero: ");
    scanf ("%d",&num);
    if (num > 10)
        printf ("\n\n O numero e maior que 10");
    else if (num == 10)
    {
        printf ("\n\n Voce acertou!\n");
        printf ("O numero e igual a 10.");
    }
    else if (num < 10)
        printf ("\n\n O numero e menor que 10");
}
```

A Expressão Condicional

- Quando o compilador avalia uma condição, ele quer um valor de retorno para poder tomar a decisão. Mas esta expressão não necessita ser uma expressão no sentido convencional. Uma variável sozinha pode ser uma "expressão" e esta retorna o seu próprio valor.

A Expressão Condicional

- Isto quer dizer que teremos as seguintes expressões:

```
int num;  
if (num!=0) ....  
if (num==0) ....  
for (i = 0; string[i] == '\0'; i++)
```

eqüivalem a

```
int num;  
if (num) ....  
if (!num) ....  
for (i = 0; string[i]; i++)
```

ifs Aninhados

- O **if** aninhado é simplesmente um **if** dentro da declaração de um outro **if** externo. O único cuidado que devemos ter é o de saber exatamente a qual **if** um determinado **else** está ligado.

Exemplo

```
#include <stdio.h>
void main ()
{
    int num;
    printf ("Digite um numero: ");
    scanf ("%d", &num);
    if (num == 10)
    {
        printf ("\n\n Voce acertou!\n");
        printf ("O numero e igual a 10.\n");
    }
    else
    {
        if (num > 10)
            printf ("O numero e maior que 10.");
        else
            printf ("O numero e menor que 10.");
    }
}
```

O Operador ?

- Uma expressão como:

```
if (a > 0)  
    b = -150;
```

```
else  
    b = 150;
```

pode ser simplificada usando-se o operador ? da seguinte maneira:

```
b = a > 0 ? -150 : 150;
```

O Operador ?

- De uma maneira geral expressões do tipo:

```
if (condição)  
    expressão_1;
```

```
else  
    expressão_2;
```

podem ser substituídas por:

```
condição ? expressão_1 : expressão_2;
```

O Operador ?

- O operador ? é limitado (não atende a uma gama muito grande de casos) mas pode ser usado para simplificar expressões complicadas. Uma aplicação interessante é a do contador circular.

O Operador ?

```
#include <stdio.h>
void main()
{
    int index = 0, contador;
    char letras[5] = "Joao";
    for (contador = 0; contador < 1000; contador++)
    {
        printf("\n%c", letras[index]);
        index = (index == 3) ? index = 0: ++index;
    }
}
```

O nome Joao é escrito na tela verticalmente até a variável contador determinar o término do programa. Enquanto isto a variável *index* assume os valores 0, 1, 2, 3, 0, 1, ... progressivamente.

O Comando switch

- O comando **switch** é próprio para se testar uma variável em relação a diversos valores pré-estabelecidos. Sua forma geral é:

```
switch (expressão)
{
    case constante_1:
        seqüência de comandos;
        break;
    case constante_2:
        seqüência de comandos;
        break;
    ...
    case constante_n:
        seqüência de comandos;
        break;
    default
        seqüência de comandos;
}
```

O Comando switch

- Podemos fazer uma analogia entre o **switch** e a estrutura **if-else-if** apresentada anteriormente.
- A diferença fundamental é que a estrutura **switch** não aceita expressões. Aceita apenas constantes.
- O **switch** testa a variável e executa a declaração cujo case corresponda ao valor atual da variável.
- A declaração **default** é opcional e será executada apenas se a variável, que está sendo testada, não for igual a nenhuma das constantes.

O Comando switch

- O comando **break**, faz com que o **switch** seja interrompido assim que uma das seqüência de comandos seja executada.
- Mas ele não é essencial ao comando **switch**. Se após a execução da declaração não houver um **break**, o programa continuará executando.
- Isto pode ser útil em algumas situações, mas eu recomendo cuidado.

Exemplo

```
#include <stdio.h>
void main ()
{
    int num;
    printf ("Digite um numero: ");
    scanf ("%d", &num);
    switch (num)
    {
        case 9:
            printf ("\n\n O numero e igual a 9.\n");
            break;
        case 10:
            printf ("\n\n O numero e igual a 10.\n");
            break;
        case 11:
            printf ("\n\n O numero e igual a 11.\n");
            break;
        default:
            printf ("\n\n O numero nao e nem 9 nem 10 nem 11.\n");
    }
}
```