

## Capítulo 3 – Comandos de Decisão

- A linguagem C oferece quatro estruturas de decisão: **if**, **if-else**, **switch** e o **operador condicional**.

### O comando if

- O comando **if** instrui o computador a tomar uma decisão simples.
- Sintaxe: **if**(expressão de teste)  
instrução;
- Se a expressão de teste for verdadeira (algo diferente de 0), o comando **if** executa a instrução. Caso contrário a instrução não será executada.
- O programa seguinte mostra o uso do comando **if**:

```
#include <stdio.h>
#include <conio.h>
main()
{
    char ch;
    ch = getche();
    if( ch == 'p')
        printf("\nVoce pressionou a tecla p.");
}
```

- Se você pressionar p o programa imprimirá "Você pressionou a tecla p". Se você pressionar qualquer outra tecla, o programa não fará absolutamente nada.

### Múltiplas instruções no corpo do comando if

- Caso múltiplas instruções sejam necessárias no corpo do comando **if** elas devem estar entre chaves.
- Vamos reescrever o programa anterior acrescentando outras instruções ao corpo do comando **if**:

```
#include <stdio.h>
#include <conio.h>
main()
{
    if( getche() == 'p') {
        printf("\nVoce pressionou a tecla p.");
        printf("\nPressione qualquer tecla para terminar.");
        getche();
    }
}
```

- No exemplo acima incluímos a função **getche()** na expressão de teste. O **if** compara o caracter retornado por **getche()** com a letra p e, se forem iguais, a expressão assume um valor verdadeiro, e o bloco de instruções é executado.

### Comandos if aninhados

- Um comando **if** pode estar dentro de outro comando **if**. Dizemos que o **if** interno está aninhado. Exemplo:

```
#include <stdio.h>
#include <conio.h>
main()
{
    char ch;
    printf("Digite uma letra entre a e z: ");
    ch = getche();
    if( ch >= 'a')
        if( ch <= 'z')
            printf("\nVoce digitou certo!");
}
```

- Este programa somente imprimirá a frase se você digitar uma letra minúscula.

### O comando if-else

- Nos exemplos anteriores o comando **if** executará a instrução (ou instruções), se a expressão de teste for verdadeira. Não fará nada se a expressão de teste for falsa.
- O comando **else**, quando associado ao **if**, executará uma instrução (ou instruções) se a expressão de teste do comando **if** for falsa.
- Sintaxe: **if**(expressão de teste)  
     instrução\_1;  
     **else**  
     instrução\_2;
- O programa seguinte imprime uma mensagem caso a expressão de teste do **if** seja falsa:

```
#include <stdio.h>
#include <conio.h>
main()
{
    char ch;
    ch = getche();
    if( ch == 'p')
        printf("\nVoce pressionou a tecla p.");
    else
        printf("\nVoce nao pressionou a tecla p.");
}
```

- O programa seguinte é uma versão muito simples do jogo de adivinhar o “número mágico”:

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
main()
{
    int magico, palpite;

    srand((unsigned) time(NULL));
    magico = rand()%10;

    printf("\nAdivinhe o numero magico [0-9]: ");
    scanf("%d", &palpite);
    if( palpite == magico )
        printf("Certo! ");
    else
        printf("Errado! ");
    printf("O numero magico eh: %d\n", magico);
}
```

- O programa produz um número mágico usando o gerador de números aleatórios **rand()**, que devolve um número arbitrário entre 0 e **RAND\_MAX**. A função **srand()** evita que **rand()** gere sempre o mesmo número aleatório, para tanto deve sempre utilizar uma *semente* diferente. **rand()**, **srand** e **RAND\_MAX** estão definidos na biblioteca **stdlib.h**.
- A função **time()** retorna a hora local em segundos e está definida na biblioteca **time.h**.

### Comandos if-else aninhados

- Utilizaremos o programa `calc.c` para exemplificar a utilização de comandos **if-else** aninhados. Este programa simula uma calculadora simples com 4 operações.

```
#include <stdio.h>
main()
{
    float num1, num2;
    char op;
    printf("\nDigite uma expressão no formato: num op num: ");
    scanf("%f %c %f", &num1, &op, &num2);
    if( op == '+' )
        printf(" = %f", num1+num2);
    else
        if( op == '-' )
            printf(" = %f", num1-num2);
        else
            if( op == '*' )
                printf(" = %f", num1*num2);
            else
                if( op == '/' )
                    printf(" = %f", num1/num2);
}
```

### O comando if-else-if

- Uma construção comum em programação é a forma **if-else-if**. Para exemplificar vamos reescrever o programa `calc.c`.

```
#include <stdio.h>
main()
{
    float num1, num2;
    char op;
    printf("\nDigite uma expressão no formato: num op num: ");
    scanf("%f %c %f", &num1, &op, &num2);
    if( op == '+' )
        printf(" = %f", num1+num2);
    else if( op == '-' )
        printf(" = %f", num1-num2);
    else if( op == '*' )
        printf(" = %f", num1*num2);
    else if( op == '/' )
        printf(" = %f", num1/num2);
}
```

- As expressões de teste são avaliadas de cima para baixo. Assim que uma expressão verdadeira é encontrada, o comando associado a ela é executada.
- Vamos utilizar o **if-else-if** para melhorar o programa do “número mágico”:

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
main()
{
    int magico, palpite;

    srand((unsigned) time(NULL));
    magico = rand()%10;
    printf("\nAdivinhe o numero magico [0-9]: ");
    scanf("%d", &palpite);
    if( palpite == magico ) {
        printf("Certo! ");
        printf("O numero magico eh: %d\n", magico);
    }
    else if( palpite > magico )
        printf("Errado, muito alto\n");
    else printf("Errado, muito baixo\n");
}
```

### Operador condicional ternário: ?

- O operador condicional ? nos dá uma maneira compacta de expressar uma instrução **if-else** simples.
- Sintaxe: condição ? expressão1 : expressão2;
- A condição é avaliada e se for verdadeira a expressão toda assume o valor da expressão1, se for falsa a expressão assume o valor da expressão2.
- Exemplo:

```
#include <stdio.h>
main()
{
    int num1, num2, max;
    printf("\nInforme dois numeros: ");
    scanf("%d %d", &num1, &num2);
    max = (num1 > num2) ? num1 : num2;
    printf("\nO maior numero eh: %d", max);
}
```

- Esta expressão é equivalente à construção **if-else** seguinte:

```
if( num1 > num2 )
    max = num1;
else
    max = num2;
```

### O comando switch

- O comando **switch** teste sucessivamente o valor de uma expressão contra uma lista de constantes inteiras ou caracteres. Quando o valor coincide, os comandos associados àquela constante são executados.
- Sintaxe: **switch**(expressão constante) {
  - case** constante1:
    - instruções; **break**;
  - case** constante2:
    - instruções; **break**;
  - default**:
    - instruções;

- O comando **break** causa uma saída imediata do **switch**. Se não existir um comando **break** seguindo as instruções de um caso, o programa segue executando todas as instruções dos casos abaixo.
- Se nenhum caso for satisfeito e existir um caso **default**, a execução começará nele, senão o programa processará as instruções seguintes ao bloco **switch**. Um **default** é opcional.
- Reescreveremos o programa `calc.c` fazendo uso do comando **switch**.

```
#include <stdio.h>
main()
{
    float num1, num2;
    char op;
    printf("\nDigite uma expressão no formato: num op num: ");
    scanf("%f %c %f", &num1, &op, &num2);
    switch( op ) {
        case '+':
            printf(" = %f", num1+num2); break;
        case '-':
            printf(" = %f", num1-num2); break;
        case '*':
            printf(" = %f", num1*num2); break;
        case '/':
            printf(" = %f", num1/num2); break;
        default:
            printf("\nOperador desconhecido");
    }
}
```

- O próximo exemplo mostrará casos sem **break**. Nesta versão você poderá digitar o sinal \* ou x para multiplicação.

```
#include <stdio.h>
main()
{
    float num1, num2;
    char op;
    printf("\nDigite uma expressão no formato: num op num: ");
    scanf("%f %c %f", &num1, &op, &num2);
    switch( op ) {
        case '+':
            printf(" = %f", num1+num2); break;
        case '-':
            printf(" = %f", num1-num2); break;
        case '*':
        case 'x':
            printf(" = %f", num1*num2); break;
        case '/':
            printf(" = %f", num1/num2); break;
        default:
            printf("\nOperador desconhecido");
    }
}
```

- O próximo programa mostra um exemplo com **switch**'s aninhados. É um banco de dados simples que mostra o desempenho dos vendedores de cada região.

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

```
main()
{
    char regioao, vendedor;
    printf("\nAs regioes sao: Leste, Oeste e Norte\n");
    printf("Informe a primeira letra da regioao: ");
    regioao = getche();

    switch( regioao )
    {
        case 'l':
            printf("\nOs vendedores sao: Rafael, Joao e Mariana\n");
            printf("Informe a primeira letra do nome do vendedor: ");
            vendedor = getche();
            switch( vendedor )
            {
                case 'r': printf("\nVendas: R$ 10.000,00\n"); break;
                case 'j': printf("\nVendas: R$ 12.000,00\n"); break;
                case 'm': printf("\nVendas: R$ 14.000,00\n");
            }
            break;
        case 'o':
            printf("\nOs vendedores sao: Ronaldo, Lisa e Hilton\n");
            printf("Informe a primeira letra do nome do vendedor: ");
            vendedor = getche();
            switch( vendedor )
            {
                case 'r': printf("\nVendas: R$ 10.000,00\n"); break;
                case 'l': printf("\nVendas: R$ 9.500,00\n"); break;
                case 'h': printf("\nVendas: R$ 13.000,00\n");
            }
            break;
        case 'n':
            printf("\nOs vendedores sao: Tomas, Jaime e Raquel\n");
            printf("Informe a primeira letra do nome do vendedor: ");
            vendedor = getche();
            switch( vendedor )
            {
                case 't': printf("\nVendas: R$ 5.000,00\n"); break;
                case 'j': printf("\nVendas: R$ 9.000,00\n"); break;
                case 'r': printf("\nVendas: R$ 14.000,00\n");
            }
        }
    }
}
```