

Estruturas de Controle de Fluxo

Estruturas Condicionais

- Permitem a escolha do grupo de ações e estruturas a serem executadas quando determinadas condições são ou não satisfeitas.

if-else-if

- Sintaxes reduzidas (um comando por decisão):

```
if (condição) comando1;
```

```
if (condição) comando1;  
else comando2;
```

```
if (condição1) comando1;  
else if (condição2) comando2;  
else if (condição3) comando3;  
:  
else comandoN;
```

- Sintaxes completas (mais de um comando por decisão):

```
if (condição1) {  
    comando1.1;  
    comando1.2;  
}
```

```
if (condição1) {  
    comando1.1;  
    comando1.2;  
}  
else {  
    comando2.1;  
    comando2.2;  
}
```

```
if (condição1) {  
    comando1.1;  
    comando1.2;  
}  
else if (condição2) {  
    comando2.1;  
    comando2.2;  
}  
else {  
    comando3.1;  
    comando3.2;  
}
```

- Se a condição da cláusula `if` for avaliada como *verdadeira* (qualquer coisa menos 0), o bloco de comandos correspondente será executado. No entanto, se a condição for *falsa*, o computador tentará avaliar a próxima condição `else-if`. Se esta cláusula não existir no código, executará o bloco de comandos da cláusula `else` (se esta existir no código).

operador ternário

- É uma maneira compacta de expressar o *if-else*
- Sintaxe (se condição for verdadeira, executa expressão1; do contrário, executa expressão2):

```
condição ? expressão1 : expressão2;
```

switch

- Uma variável é comparada com uma lista de variáveis numéricas ou alfanuméricas. Encontrando uma equivalência, o bloco de comandos correspondente é executado.
- Uso mais comum na construção de menus.
- Sintaxe:

```
switch (variável) {  
    case constante1:  
        seqüencia de comandos 1  
        break;  
    case constante2:  
        seqüencia de comandos 2  
        break;  
    :  
    default:  
        seqüencia de comandos 3  
        break;  
}
```

Exercício: Fazer um programa em linguagem C para cada um dos seguintes problemas:

- a) *Maior* – Receber dois números reais e imprimir na tela o maior entre eles. Se ambos forem iguais, deve ser exibida uma mensagem na tela.

```
#include <stdio.h>  
int main() {  
    int a, b;  
  
    scanf("%d %d", &a, &b);  
    if (a > b)  
        printf("%d", a);  
    else if (a < b)  
        printf("%d", b);  
    else  
        printf("Numeros iguais");  
  
    return 0;  
}
```

- b) *Calculadora Manual* – Receber um número real, um sinal aritmético (“+”, “-”, “/”, “*”) e outro número real e executar a operação desejada, imprimindo o resultado no monitor. O programa deve ser capaz de verificar operações inválidas (no caso, divisão por zero), exibindo uma mensagem de alerta.

```
#include <stdio.h>

int main()
{
    float a, b;
    char operador;

    scanf("%f %c %f", &a, &operador, &b);
    if (operador == '+')
        printf( "%f", (a + b) );
    else if (operador == '-')
        printf( "%f", (a - b) );
    else if (operador == '*')
        printf( "%f", (a * b) );
    else if (operador == '/') {
        if (b != 0)
            printf( "%f", (a / b) );
        else
            printf("Divisao por zero nao existe");
    }
    else
        printf("Operador desconhecido");

    return 0;
}
```

- c) *Ecoar um caractere* – Receber um caractere qualquer do usuário via teclado e o imprimir no monitor. Se o caractere digitado for uma letra minúscula, ele deve ser ecoado como letra maiúscula. Dica: verifique os códigos de letras na tabela ASCII (a função `toupper ()` não deve ser usada).

```
#include <stdio.h>

int main()
{
    char caractere;
    int codigo;

    scanf("%c", &caractere);
    codigo = caractere;
    if ((codigo >= 97) && (codigo <= 122))
        codigo = codigo - 32;
    printf("%c", codigo);

    return 0;
}
```

- d) *Relógio* – Receber dois horários do usuário via teclado, cada um com três variáveis (hora, minuto, segundo) e imprimir a soma deles. No resultado, não devem aparecer minutos ou segundos maiores que 59.

```
#include <stdio.h>

int main()
{
    int hora1, min1, seg1;
    int hora2, min2, seg2;
    int hora = 0;
    int min = 0;
    int seg = 0;

    scanf("%d %d %d", &hora1, &min1, &seg1);
    scanf("%d %d %d", &hora2, &min2, &seg2);
    seg = seg1 + seg2;
    if (seg > 59) {
        min = seg / 60;
        seg = seg % 60;
    }
    min = min + min1 + min2;
    if (min > 59) {
        hora = min / 60;
        min = min % 60;
    }

    hora = hora + hora1 + hora2;
    printf("%d %d %d", hora, min, seg);

    return 0;
}
```

- e) *Menu* – Criar um programa que exemplifica o uso do `switch` através da escolha de itens de um menu. Dadas algumas opções, o usuário escolhe uma delas e o programa toma a decisão correspondente.
- f) *Checação de letras* – Receber dois caracteres do usuário via teclado e verificar se ambos são letras. Dica: verifique os códigos de letras na tabela ASCII.
- g) *Regiões de um plano* – Receber dois números reais correspondentes às duas coordenadas x e y de um ponto no plano e verificar se ele pertence a uma das seguintes regiões, às duas ou à nenhuma delas:
- h) *Equação do segundo grau* – Receber três números reais do usuário via teclado correspondentes aos termos independentes a , b e c de uma equação do segundo grau: $ax^2 + bx + c = 0$. Em seguida, calcular as raízes da equação e imprimi-las no monitor. Lembre-se: uma equação do segundo grau pode ter duas, uma ou nenhuma raiz. Você vai precisar usar as seguintes fórmulas:

a. $\text{delta} = b^2 - 4ac$

b. $\text{raízes} = \frac{-b \pm \sqrt{\text{delta}}}{2a}$