



Laboratório de Informática

Estrutura de controle Aula 11

Profª Ms. Lília Marta Brandão Soussa Modesto

Desvio Condicional Simples

```
if (<condição>)  
    <instrução para condição verdadeira>;  
<instrução para condição falsa ou após ser verdadeira>;
```

Exemplo:

```
if (X != 10)  
    printf ("\n Valor diferente de 10");  
Y = X * 3;
```

```
if (<condição>)  
{  
    <instrução 1 para condição verdadeira>;  
    <instrução 2 para condição verdadeira>;  
    <instrução 3 para condição verdadeira>;  
    <instrução n para condição verdadeira>;  
}
```

Exemplo:

```
if (A > B)  
{  
    X = A;  
    A = B;  
    B = X;  
}  
Z = A / 2;
```

```
<instrução para condição falsa ou após ser verdadeira>;
```

Desvio Condicional Composto

```
if (<condição>)  
    <instrução para condição verdadeira>;  
else  
    <instrução para condição falsa>;
```

Exemplo:

```
if (X == 5)  
    Y = X * 3;  
else  
    Y = X + 5;
```

```
if (<condição>)  
{  
    <instrução 1 para condição verdadeira>;  
    <instrução 2 para condição verdadeira>;  
    <instrução n para condição verdadeira>;  
}  
else  
{  
    <instrução 1 para condição falsa>;  
    <instrução 2 para condição falsa>;  
    <instrução n para condição falsa>;  
}
```

Exemplo:

```
if (X == 5)  
{  
    Y = X * 3;  
    Z = X;  
    X = Y + Z;  
}  
else  
{  
    Y = X + 5;  
    Z = X - 2;  
}
```

Desvio Condicional Encadeado

Exemplo: Crie um programa que efetue o cálculo do reajuste de salário, segundo a tabela abaixo:

Salário	Reajuste
< 500	15%
≥ 500 e ≤ 1000	10%
> 1000	5%

Algoritmo:

- 1- Ler um valor para a variável SAL;
- 2- Verificar se o valor de SAL < 500, se sim reajustar em 15%;
- 3- Caso contrário, verificar se o valor de SAL ≤ 1000 , se sim reajustar em 10%;
- 4- Caso contrário, reajustar SAL em 5%;
- 5- Apresentar o valor da variável SAL.

Desvio Condicional Encadeado

```
/* Programa Reajusta Salário */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{ system("color f0"); float SAL;
  printf ("\nDigite o Salário: ");
  scanf ("%f", &SAL);
  if (SAL < 500)
    SAL = SAL * 1.15;
  else
    if (SAL <= 1000)
      SAL = SAL * 1.10;
    else
      SAL = SAL * 1.05;
  printf ("\nSalário = %.2f\n", SAL);
  system("pause");
  return(0); }
```

Operador Lógico .e.

Este operador é utilizado quando todas as condições necessitam ser verdadeiras.

Linguagem C:

```
if (<condição1> && <condição2>)  
    <instrução para condição1 e condição2  
    verdadeiras>;
```

Exemplo:

```
if (X >= 0 && X <= 9)  
    printf ("\nValor válido");
```

Operador Lógico .ou.

Este operador é utilizado quando pelo menos uma condição necessita ser verdadeira.

Linguagem C:

```
if (<condição1> || <condição2>)  
    <instrução para condição1 ou condição2  
    verdadeiras>;
```

Exemplo:

```
if (X == 'F' || X == 'M')  
    printf ("\nOpção válida");
```

Operador Lógico .não.

Este operador é utilizado quando for necessário inverter o resultado lógico da condição.

Linguagem C:

```
if ( ! ( <condição> ))  
    <instrução para condição não verdadeira>;
```

Exemplo:

```
if ( ! ( X > 5 ))  
    printf ("\nValor menor que 5");
```


Exemplo

Crie um programa que leia três valores para os lados de um triângulo, considerando lados como: A, B e C. Verifique se os lados formam realmente um triângulo, se sim, apresente qual tipo de triângulo é: equilátero, isósceles ou escaleno.

Obs: É triângulo quando cada lado é menor que a soma dos outros dois lados. Então: $A < B + C$, $B < A + C$ e $C < A + B$.

Equilátero: possui todos os lados iguais: $A = B$ e $B = C$

Isósceles: possui 2 lados iguais e 1 diferente.

$(A = B \text{ e } A \neq C)$ ou $(A = C \text{ e } A \neq B)$ ou $(B = C \text{ e } B \neq A)$

Escaleno: possui todos os lados diferentes.

$A \neq B$, $A \neq C$ e $B \neq C$.

Exemplo

```
/* Programa TRIANGULO */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    system("color f0");
    float A, B, C;
    printf ("\n Digite os 3 lados: ");
    scanf ("%f %f %f", &A, &B, &C);
    if (A<B+C && B<A+C && C<A+B)
        if (A==B && B==C)
            printf ("\n Triangulo Equilátero\n");
        else
            if (A==B || A==C || B==C)
                printf ("\n Triangulo Isosceles\n");
            else
                printf ("\n Triangulo Escaleno\n");
    else
        printf ("\n Nao eh um triangulo\n");
    system("pause");    return(0);
}
```

Estrutura de Controle com Múltipla Escolha

Sintaxe:

```
switch <variável>
{
    case <opção 1> : <operação 1>; break;
    case <opção 2> : <operação 2>; break;
    ...
    case <opção n> : <operação n>; break;
    default       : <operação>; break;
}
```

Exemplo: Faça um programa que simule uma calculadora com as 4 operações básicas (Adição / Subtração / Multiplicação / Divisão).

Estrutura de Controle com Múltipla Escolha

```
/* Programa Calculadora */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    system("color f0");
    float N1, N2, RES;
    char OP;
    printf ("\n Calculadora das 4 operacoes basicas \n");
    printf ("\n Digite um numero: ");
    scanf ("%f", &N1);
    printf ("\n Digite outro numero: ");
    scanf ("%f", &N2);
    printf ("\n Digite a operacao [+ - / *]: ");
    fflush(stdin);
    OP = getchar();
```

Estrutura de Controle com Múltipla Escolha

```
switch (OP)
```

```
{
```

```
    case '+' : RES = N1 + N2;          break;
```

```
    case '-' : RES = N1 - N2;          break;
```

```
    case '*' : RES = N1 * N2;          break;
```

```
    case '/' : if (N2 != 0)
                  RES = N1 / N2;
```

```
        else
```

```
            printf ("\n Divisao por ZERO");
```

```
            break;
```

```
    default : printf ("\n Opcao Invalida");break;
```

```
}
```

```
if (OP == '+' || OP == '-' || OP == '*' || OP == '/')
```

```
    printf ("\n %.1f %c %.1f = %.1f \n\n",N1,OP,N2,RES);
```

```
system("pause");
```

```
return(0);
```