

ESTRUTURAS CONDICIONAIS



Baseado nos slides de autoria de Rosely Sanches e Simone Senger de Souza

Estruturas de Controle

• ESTRUTURA SEQUENCIAL

• ESTRUTURA CONDICIONAL

• ESTRUTURA DE REPETIÇÃO

Estruturas Condicionais

- Estrutura Condicional Simples
- Estrutura Condicional Composta
- Seleção entre duas ou mais Sequências de Comandos

Estrutura Condicional Simples

Formas de Representação no Algoritmo

Algoritmo

se (condição)

então <comando>

fim-se

Estrutura Condicional Simples C

```
if (condição)
```

```
<comando> ;
```

Estrutura C

a condição deve ser uma
expressão lógica

```
if (condição)
```

```
<comando> ;
```

O comando só será executado
se a condição for verdadeira

Estrutura Condicional Simples C

if (condição)

<comando>

se **mais de um comando** deve ser executado quando a **condição** for verdadeira, esses comandos devem ser transformados em um **comando composto**.

Comando Composto

- Um conjunto de comandos que devem ser executados em uma ordem específica.
- Os comandos devem ficar reunidos entre as palavras chaves { e }

• {
 comando 1 ;
 comando 2 ;
 comando 3 ;
}

Comando composto

A diagram illustrating a composite command. On the left, a pink curly brace groups three lines of text: 'comando 1 ;', 'comando 2 ;', and 'comando 3 ;'. To the right of this group, a larger black curly brace groups the entire set of three lines, and the text 'Comando composto' is written in blue to the right of this larger brace.

Estrutura Condicional Simples C

```
if (condição)
{
    <comando> ;
    <comando> ;
    <comando> ;
}
```

EXEMPLO (condicional simples): Ler dois números inteiros e encontrar o maior deles

Algoritmo

Variaveis

A,B, MAIOR: inteiro;

Inicio

leia (A, B)

MAIOR = A

se (B > A)

então MAIOR = B

fim-se

escreva (MAIOR)

Fim.

Estrutura Condicional Composta

Formas de Representação no Algoritmo

Algoritmo

se (condição)

então <comando1>

senão <comando2>

fim-se

Estrutura Condicional Composta C

```
if (condição)
    <comando 1> ;
else <comando 2>;
```

Estrutura Condicional Composta C

a condição deve ser uma
expressão lógica

```
if (condição)
    <comando 1> ;
else <comando 2>;
```

Estrutura Condicional Composta C

Se condição for verdadeira será executado o comando 1 e não será executado o comando 2.

```
if (condição)
    <comando 1> ;
else <comando 2>;
```

Estrutura Condicional Composta C

Se condição for **falsa** será executado o **comando 2** e não será executado o **comando 1**.

```
if (condição)  
    <comando 1> ;  
else <comando 2>;
```

Estrutura Condicional Composta C

```
if (condição)
    <comando 1> ;
else <comando 2>;
```

se mais de um comando deve ser executado quando a condição for verdadeira ou quando a condição for falsa, esses comandos devem ser transformados em comandos compostos.

Estrutura Condicional Composta C

```
if (condição)
{
    <comando> ;
    <comando> ;
    <comando> ;
}
else
{
    <comando> ;
    <comando> ;
}
```

EXEMPLO (condicional composto): Calcular a área de um triângulo dado o comprimento de seus lados

Algoritmo area

Variáveis

A,B,C,P,AREA: real;

Início

leia (A, B, C)

se (A<B+C) e (B<A+C) e (C<B+A)

então

início

$$P = (A+B+C)/2$$

$$AREA = \sqrt{P(P-A)(P-B)(P-C)}$$

escreva (AREA)

fim

senão escreva (“Os numeros A, B e C nao formam um triangulo”)

fim-se

Fim.

Verificação se os lados formam um triângulo

Cálculo da Área

SELEÇÃO ENTRE DUAS OU MAIS SEQÜÊNCIAS DE COMANDOS -

Formas de Representação no Algoritmo

caso <valor da expressão>

igual <lista de valores 1> então <sequência de comandos 1>

.....

igual <lista de valores N> então <sequência de comandos N>

senão <sequência de comandos X>

fim-caso

SELEÇÃO ENTRE DUAS OU MAIS SEQÜÊNCIAS DE COMANDOS C

```
switch (variável) {  
    case <constante1> : <seqüência de comandos 1> ; break;  
    ....  
    case <constante N> : <seqüência de comandos N> ; break;  
    default: seqüência de comandos default;  
}
```

SELEÇÃO ENTRE UM OU MAIS SEQUÊNCIAS C

A expressão é **avaliada** e então a seqüência associada ao seu **valor** é executada.

```
switch (variável) {  
    case <constante1> : <seqüência de comandos 1> ; break;  
    ....  
    case <constante N> : <seqüência de comandos N> ; break;  
    default: seqüência de comandos default;  
}
```

Se o comando **break** não for utilizado os demais **cases** continuam sendo avaliados.

SELEÇÃO ENTRE DUAS OU MAIS SEQÜÊNCIAS DE COMANDOS C

Se o valor da expressão **não** se encontra em nenhuma das listas de valores, a **seqüência default** será executada.

```
switch (va  
  case <...> : <seqüência de comandos 1> ; break;  
  ....  
  case <constante N> : <seqüência de comandos N> ; break;  
  default: seqüência de comandos default;  
}
```

A opção **default** é facultativa

EXEMPLO (comando case) : Dado o código da peça, determinar seu peso

Algoritmo

Variaveis

CODIGO: caractere;

PESO: real;

Início

// solicitar o fornecimento do codigo da peca (P,M,G)

leia (CODIGO)

caso CODIGO

igual 'p','P' então

PESO=23.5

escreva (PESO)

igual 'm','M' então

PESO=50.7

escreva (PESO)

igual 'g','G' então

PESO=102.9

escreva (PESO)

senão escreva (“Código Invalido”)

fim-caso

Fim

EXEMPLO (comando switch) : Dado o código da peça, determinar seu peso

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[])
{
    char codigo;
    float peso;
    printf("Entre com o código do produto (P,M,G):" );
    scanf("%c", &codigo);
    switch (codigo) {
        case 'P': peso=23.5;
                printf("o peso eh %.1f\n",peso);
                break;
        case 'M': peso=50.7;
                printf("o peso eh %.1f\n",peso);
                break;
        case 'G': peso=102.9;
                printf("o peso eh %.1f\n",peso);
                break;
        default: printf("\n\ncodigo invalido! \n");
    } // fim do switch
    system("PAUSE");
    return 0;
}
```


Exemplo

1. Faça um algoritmo que lê dois números inteiros e mostra-os em ordem crescente.

EXEMPLO (condicional simples): Ler dois números inteiros e colocá-los em ordem crescente

Algoritmo ordena

Variaveis

A, B, AUX: inteiro

Inicio

leia (A, B)

se (A > B)

então

inicio

AUX = A

A = B

B = AUX

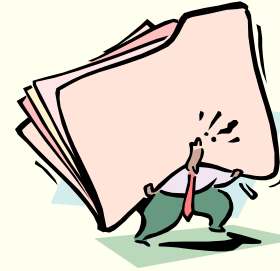
fim

fim-se

escreva (A,B)

Fim.

Exercícios



1. Elaborar um algoritmo que, dada a idade de um nadador, classificá-lo nas categorias: infantil A (5 - 7 anos), infantil B (8 -10 anos), juvenil A (11 - 13 anos), juvenil B (14 -17 anos) e adulto (maiores que 18 anos).
2. Escreva um algoritmo que lê um valor em reais e calcula qual o menor número possível de notas de 100, 50, 10, 5 e 1 em que o valor lido pode ser decomposto e escreva o valor lido e a relação de notas necessárias. Ex. R\$ 477,00 -> 4 notas de 100,00, 1 nota de 50,00, 2 notas de 10,00, 1 nota de 5,00 e 2 notas de 1,00.
3. Qualquer número natural de quatro algarismos pode ser dividido em duas dezenas formadas pelos seus dois primeiros e dois últimos dígitos. (1297 = 12 e 97; 5314 = 53 e 14). Escreva um algoritmo que lê um número inteiro n (de 4 algarismos) e verifica se a raiz quadrada de n é igual a soma das dezenas de n .
Ex.: $n = 9801$, dezenas de $n = 98 + 01$, soma das dezenas 99, raiz quadrada de $n = 99$. Portanto a raiz quadrada de 9801 é igual a soma de suas dezenas.