

Estruturas de Controle

- ESTRUTURA SEQUENCIAL
- ESTRUTURA CONDICIONAL
- ESTRUTURA DE REPETIÇÃO

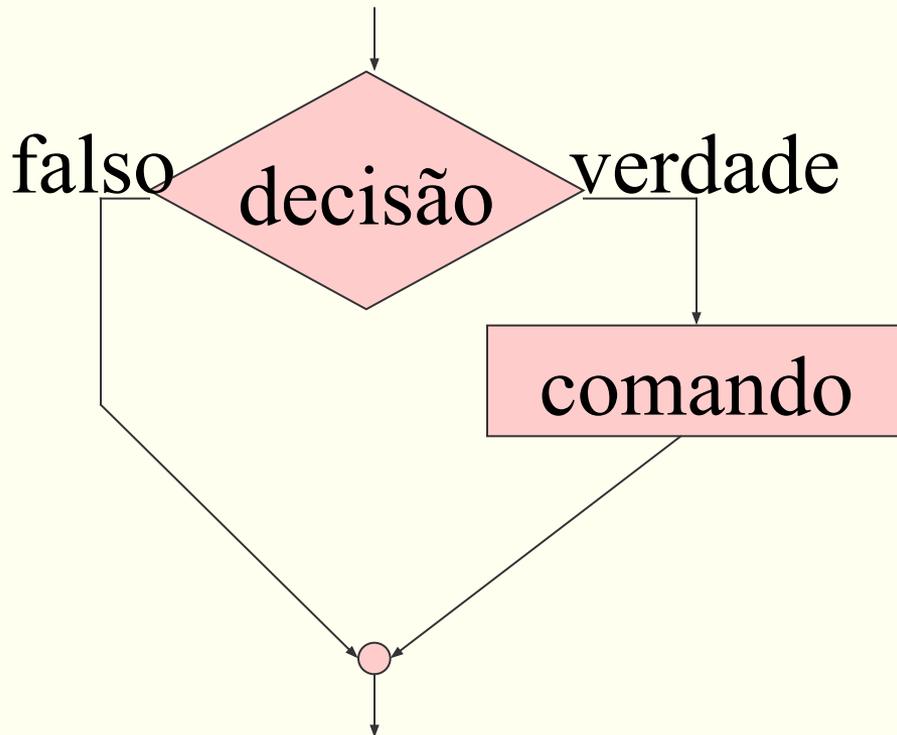
Estruturas Condicionais

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

Estrutura Condicional Simples

Formas de Representação no Algoritmo

Fluxograma



Linguagem C

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

Estrutura Condicional Simples C

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

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

Estrutura Condiciona

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



Estrutura Condicional Simples C

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

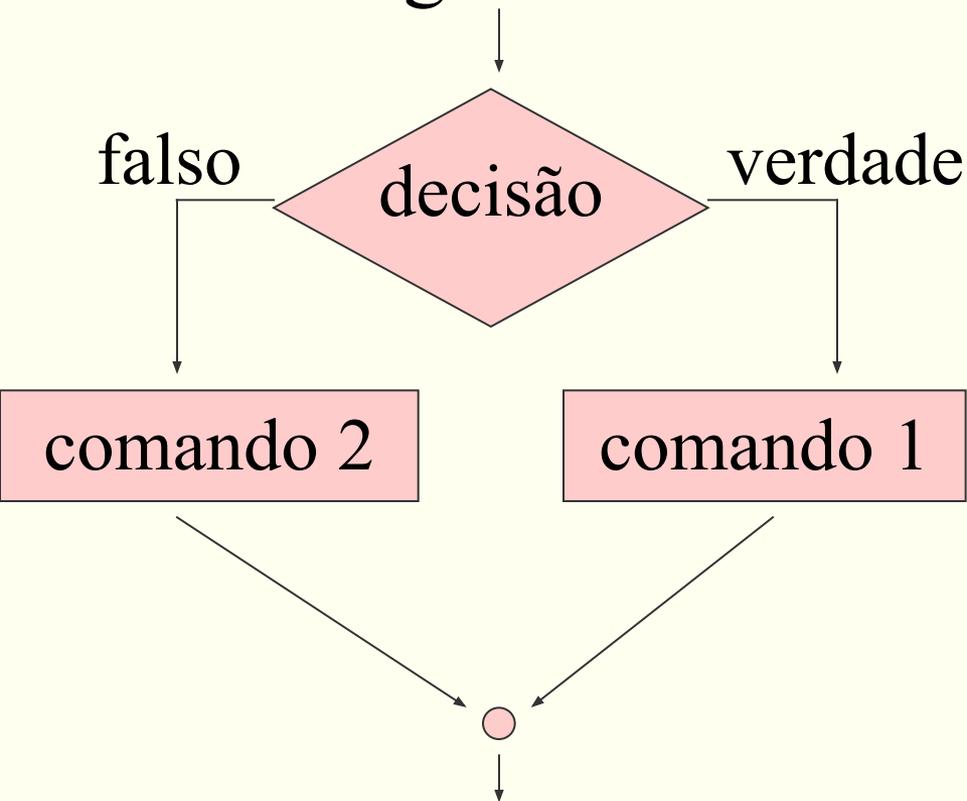
EXEMPLO (condicional simples): Ler dois números inteiros e encontrar o maior deles

```
1  #include<stdlib.h>
2  #include<stdio.h>
3
4  int main(){
5
6      int a, b, maior;
7      //Entrada de dados
8      printf("Digite dois números inteiros:");
9      scanf("%d %d",&a,&b);
10     maior = a;
11     //Estrutura condicional
12     if(b > a)
13         maior = b;
14     //Saída de dados
15     printf("Maior: %d",maior);
16
17     return 0;
18 }
19
```

Estrutura Condicional Composta

Formas de Representação no Algoritmo

Fluxograma



Linguagem C

```
if (condição) {  
    <comando1>  
}  
  
else {  
    <comando2>  
}
```

Estrutura Condicional Composta C

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

```
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

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

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> ;
}
```

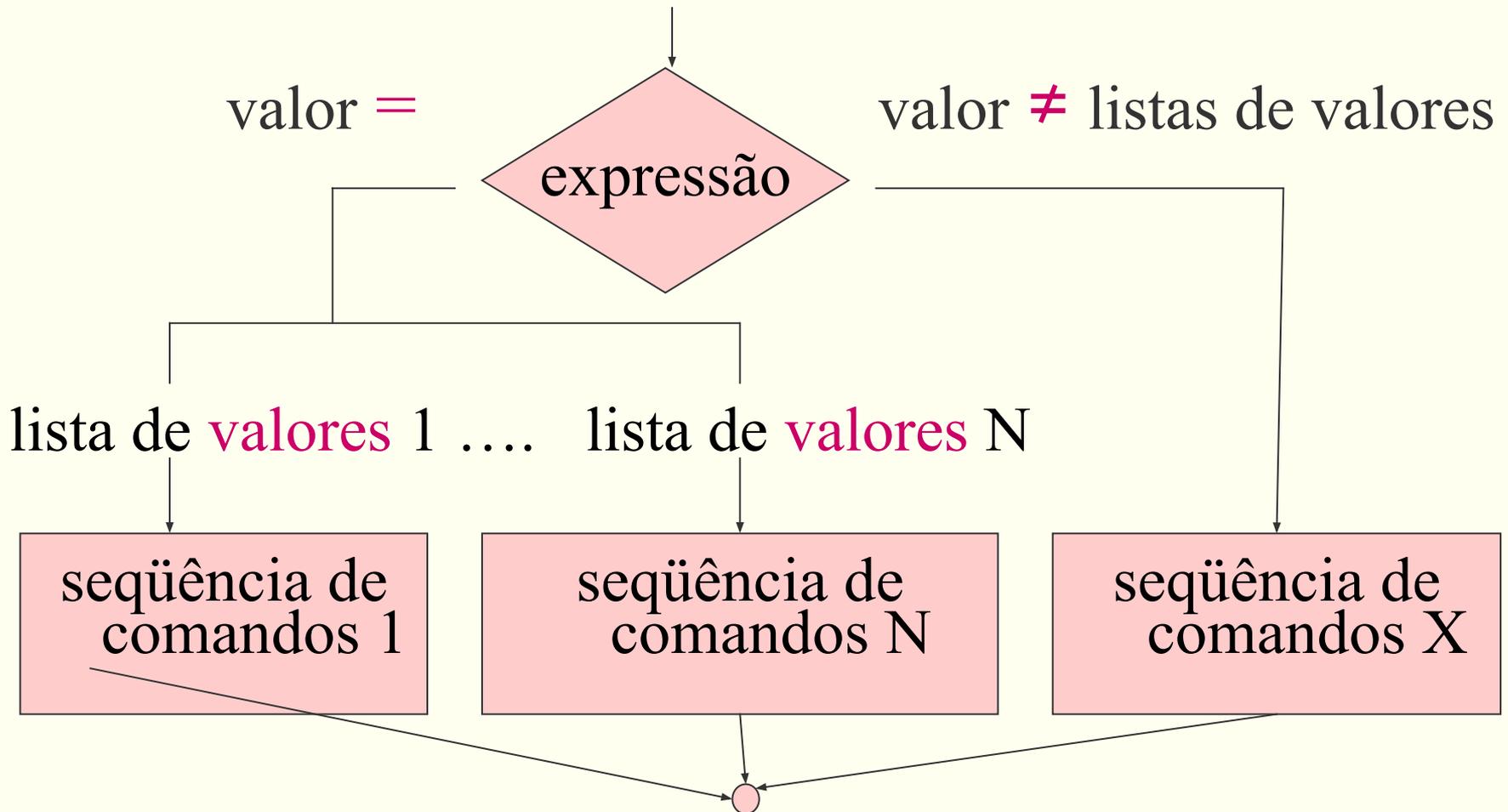
EXEMPLO (condicional composto): Calcular a área de um triângulo dado o comprimento de seus lados

```
1  #include<stdlib.h>
2  #include<stdio.h>
3  #include<math.h>
4
5  int main(){
6  float a,b,c,p,area;
7  //Entrada de dados
8  printf("Entre os lados de um triângulo:");
9  scanf("%f %f %f",&a,&b,&c);
10 //Estrutura condicional composta
11 if((a<b+c)&&(b<a+c)&&(c<b+a)){
12     p = (a+b+c)/2;
13     area = sqrt(p*(p-a)*(p-b)*(p-c));
14     printf("Area = %.3f",area);
15 }
16 else{
17     printf("Os lados %.2f %.2f %.2f não formam um triângulo",a,b,c);
18 }
19
20 return 0;
21
22 }
23
24
```

SELEÇÃO ENTRE DUAS OU MAIS SEQÜÊNCIAS DE COMANDOS

Formas de Representação no Algoritmo

Fluxograma



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;  
}
```

SELE
SEQ

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

MAIS
S C

switch (**variável**)

case <constante 1> : <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

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 : Dado o código da peça, determinar seu peso

```
1 #include<stdlib.h>
2 #include<stdio.h>
3 #include<ctype.h>
4
5 int main(){
6     char codigo;
7     float peso;
8     //Entrada de dados
9     printf("Entre código da peça (P,M,G):");
10    codigo = getchar();
11    codigo = toupper(codigo);|
12    //Estrutura condicional
13    switch(codigo){
14        case 'P': peso = 23.5;
15                 printf("%.1f", peso);
16                 break;
17        case 'M': peso = 50.7;
18                 printf("%.1f", peso);
19                 break;
20
21        case 'G': peso = 102.9;
22                 printf("%.1f", peso);
23                 break;
24
25        default: printf("Registro inválido");
26
27    }
28
29    return 0;
30 }
```

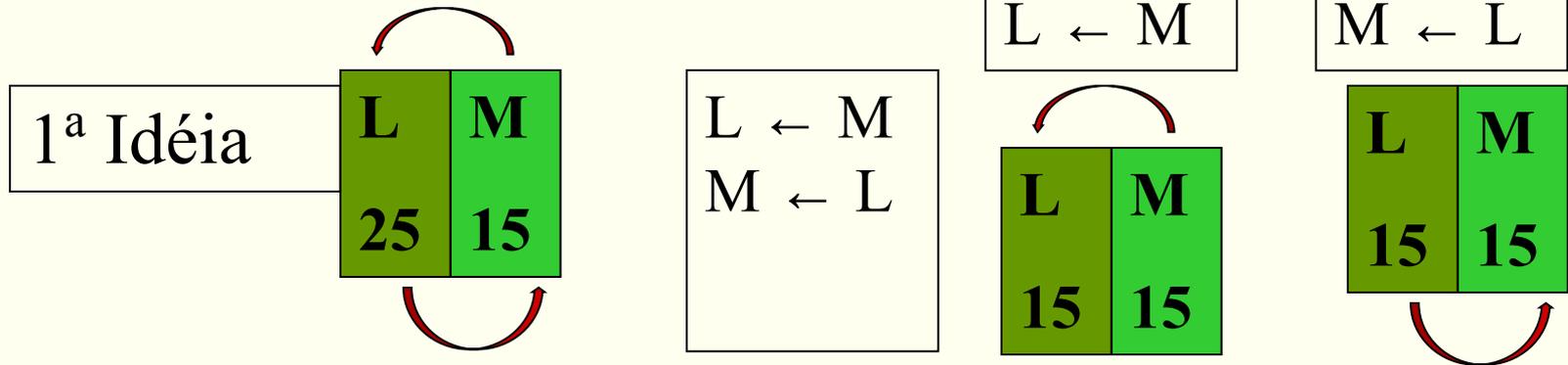
Exemplo

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

TROCA DE CONTEÚDO ENTRE DUAS VARIÁVEIS

L	M
25	15

memória



NÃO FUNCIONA

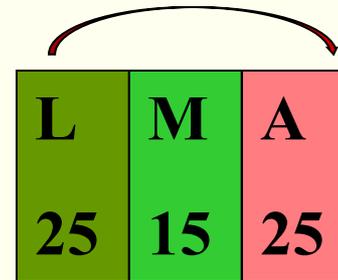
para trocar é preciso uma variável auxiliar

L	M	A
25	15	

memória

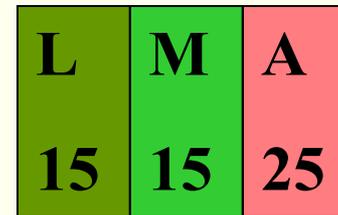
TROCA DE CONTEÚDO ENTRE DUAS VARIÁVEIS

Faço uma cópia de L em A



$A \leftarrow L$

Coloco M no lugar de L



$L \leftarrow M$

Coloco o L que estava guardado em A no lugar de M



$M \leftarrow A$

TROCA DE CONTEÚDO ENTRE DUAS VARIÁVEIS

trocar de lugar L e M

$A \leftarrow L$

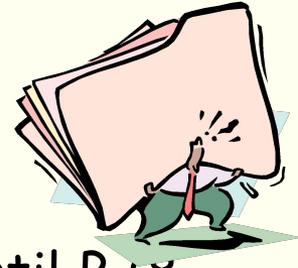
$L \leftarrow M$

$M \leftarrow A$

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

```
1  #include<stdlib.h>
2  #include<stdio.h>
3
4  int main(){
5      int a,b,aux;
6
7      printf("Entre valores inteiros:");
8      scanf("%d %d",&a,&b);
9
10     if(a>b){
11         aux = a;
12         a=b;
13         b=aux;
14     }
15     printf("Valores ordenados: <%d,%d>|", a, b);
16
17     return 0;
18
19 }
20
```

Exercícios



1. Elaborar um algoritmo que, dada a idade de um nadador, classifi-cá-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.