



Algoritmos com seleção

Inf01202 – Algoritmos e Programação

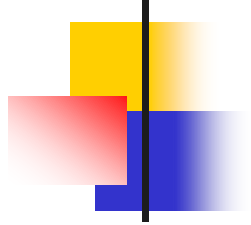
Turma: H – Modalidade EAD – 2008/1

Professoras responsáveis:

Maria Aparecida C. Livi

Maria Aparecida M. Souto

Magda Bercht



Algoritmo com seleção

Algoritmo em que a execução de determinados passos ou instruções está subordinada a uma ***condição*** !



Motivação

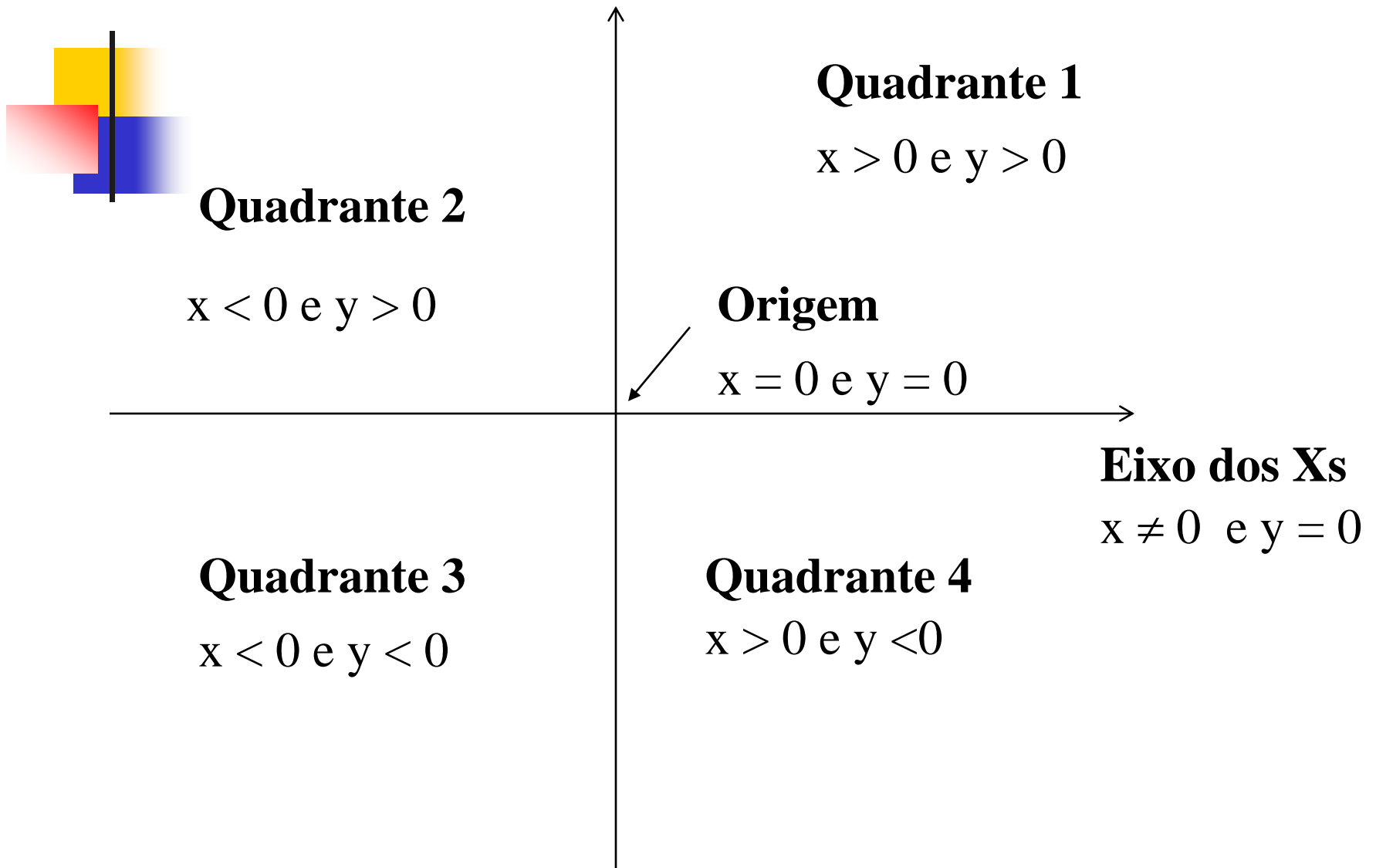
Problema1:

Localização de um ponto no plano.

Dados um par de valores x e y , que representam as coordenadas de um ponto no plano, determinar a localização do ponto: se em um quadrante, em um dos eixos ou na origem.

Saída: mensagem adequada

- **Entradas:** coordenadas x e y de um ponto
- **Processamento:**
série de testes, verificando em qual caso o par de valores enquadra-se.





Passos do Algoritmo

- Leitura ou entrada dos valores de x e y .
- Determinação, pela avaliação de condições, de onde o ponto se encontra: se em um quadrante, em um eixo ou na origem.
- Escrita da mensagem, conforme resultado da condição, onde é indicada a localização do ponto.

Escrita da condição em pseudolinguagem e em fluxograma

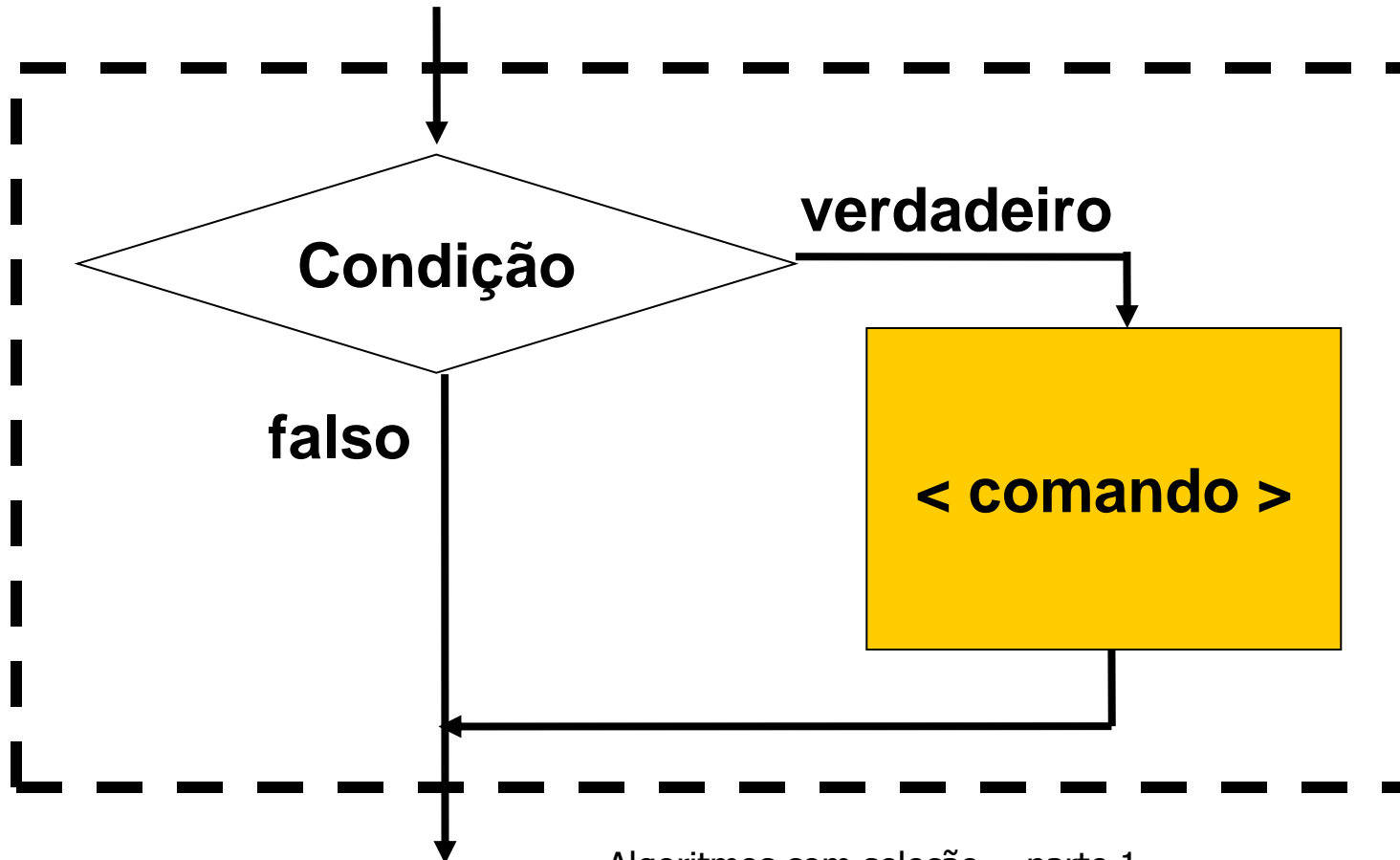
Pseudolinguagem

se $x = 0$ e $y = 0$

escrever "Ponto na origem"

se condição
comando

Fluxograma



Escrita do algoritmo em pseudolinguagem:



Início

Ler x , y

Se $x = 0$ e $y = 0$ escrever "Ponto na origem"

Se $x > 0$ e $y > 0$ escrever "Quadrante 1"

Se $x < 0$ e $y > 0$ escrever "Quadrante 2"

Se $x < 0$ e $y < 0$ escrever "Quadrante 3"

se $x > 0$ e $y < 0$ escrever "Quadrante 4"

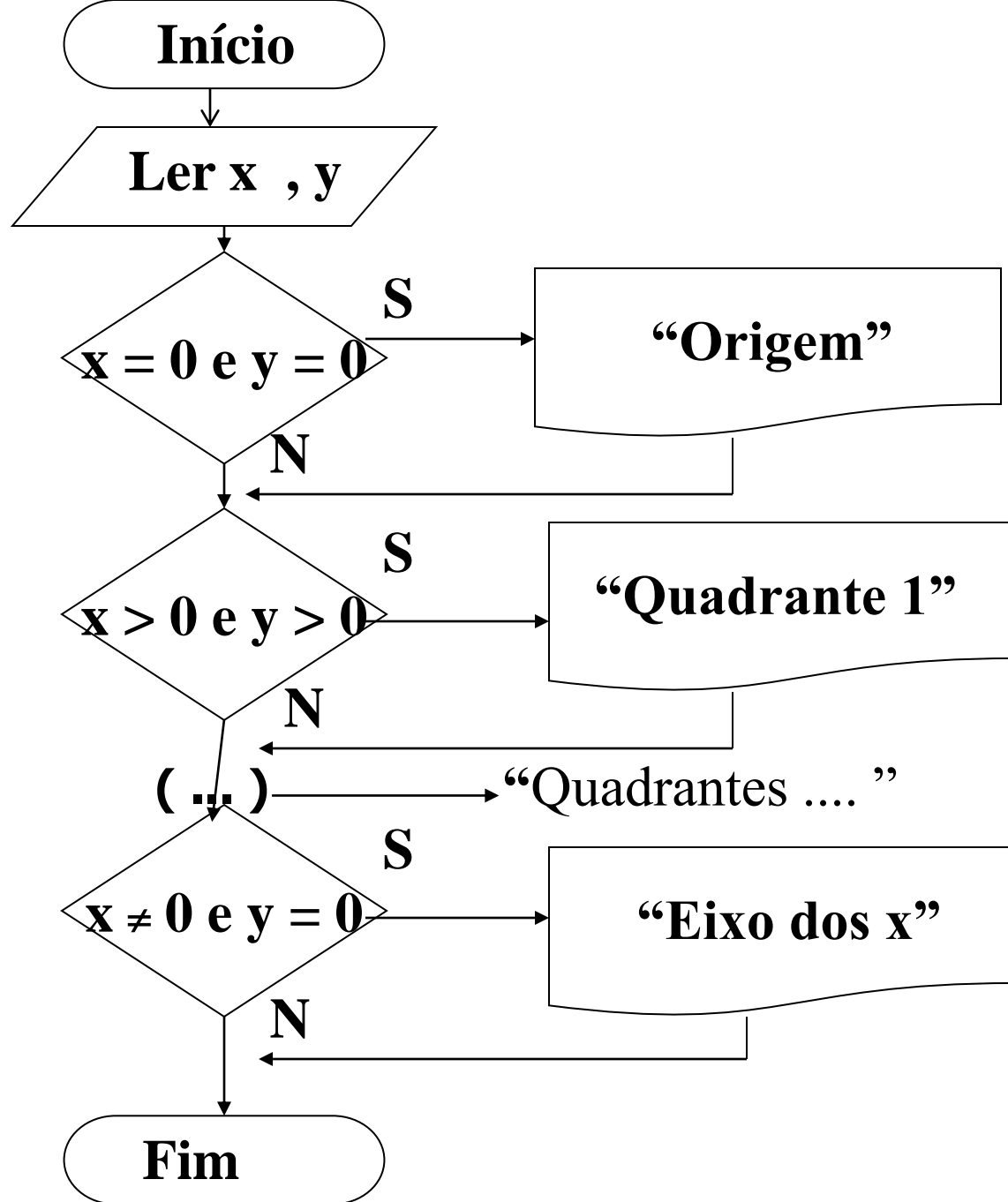
Se $x \neq 0$ e $y = 0$ escrever "Eixo dos x "

Fim

Atenção: para cada par de valores uma e apenas uma mensagem deve ser produzida.



Escrita do algoritmo em fluxograma





Implementação de algoritmos com **seleção** em linguagem C:

- Além das instruções já vistas:
 - scanf
 - printf
 - atribuição --> =

Novo comando:

if



Algoritmo com seleção ou condicionais:

- Algoritmo com seleção é aquele em que a execução de determinados passos ou instruções está subordinada a uma **condição** !
- O que é uma **condição** em C?
 - Uma condição é uma expressão cujo resultado é **verdadeiro** ou **falso**.
- O comando ***if ...*** implementa uma estrutura de seleção através da avaliação de uma **condição** conforme veremos a seguir.



Características de uma *condição* em C:

- Em C, os valores falso e verdadeiro são representados por zero e não zero. O **zero** significa **falso** e qualquer **valor diferente de zero** significa **verdadeiro**.
- As condições em C podem envolver operadores aritméticos, relacionais e lógicos.
- Se a condição for uma expressão aritmética, o próprio resultado da expressão determinará a veracidade ou falsidade da expressão.
- Se a condição envolver operadores relacionais e lógicos (como no caso da determinação do ponto no plano), o sistema avaliará a expressão e retornará um valor zero ou não zero, dependendo da falsidade ou verdade da expressão, no momento.

Exemplos de condições em C:

No problema da localização do ponto no plano, temos condições lógicas compostas, com o operador lógico **&&** conectando expressões relacionais

$(x == 0 \ \&\& \ Y < 0).$

Outros exemplos de condições em comandos *if*:

if (valor) ...

se valor for **zero**, a condição resultará **falsa**, caso contrário, **verdadeira**.

if (numero % 2) ...

se número dividido por 2 tiver por resto zero, a condição resultará **falsa**, caso contrário, **verdadeira**.

if (temperatura < 0)

se temperatura for menor que zero, o resultado da expressão é verdadeiro, mas se temperatura for igual ou maior que zero, a condição resultará falsa.



Operadores usados em expressões relacionais e lógicas:

Uma **condição** pode ser expressa através de expressões *relacionais* e/ou *lógicas*. Essas expressões são descritas, respectivamente, através do uso dos operadores:

- **relacionais**
- e
- **lógicos.**



OPERADORES RELACIONAIS:

$<$ (menor)

$>$ (maior)

$<=$ (menor ou igual)

$>=$ (maior ou igual)

$==$ (igual)

$!=$ (diferente)



Exemplos de Expressões Relacionais:

A é maior que 10?

Em C:

$(a > 10)$

X é negativo?

Em C:

$(x < 0)$

y é diferente de 10?

Em C:

$(y \neq 10)$

nota é igual a 5?

Em C:

$(nota == 5)$



OPERADORES LÓGICOS:

- **E** lógico: &&
- **Ou** lógico: ||
- **Negação**: !



Expressões Lógicas

Conectam *expressões relacionais* através de operadores lógicos.

Ex.:

$(x == 0 \ \&\& \ y == 0)$

$((k \leq 0 \ || \ k > 100) \ \&\& \ k \neq -1)$

Exemplos de expressões lógicas em C:

a) **codigo** entre 1 e 100 (valores limites considerados):

(codigo > 0 && codigo < 101)

~~**(0 < codigo < 101)**~~

Atenção:

Essas duas formas de escrever a condição não são equivalentes.

Apenas a primeira deve ser usada em casos semelhantes.

b) **contador** maior ou igual a **maximo** ou **valor** igual a **procurado**:

(contador >= maximo || valor == procurado)

c) **valor** igual a 1 ou 2 ou entre 5 e 7 (inclusive):

(valor == 1 || valor == 2 || (valor >=5 && valor <=7))



O comando *if...*

Estrutura de seleção simples

■ Comando **if**

```
if (condição)  
    comando;
```

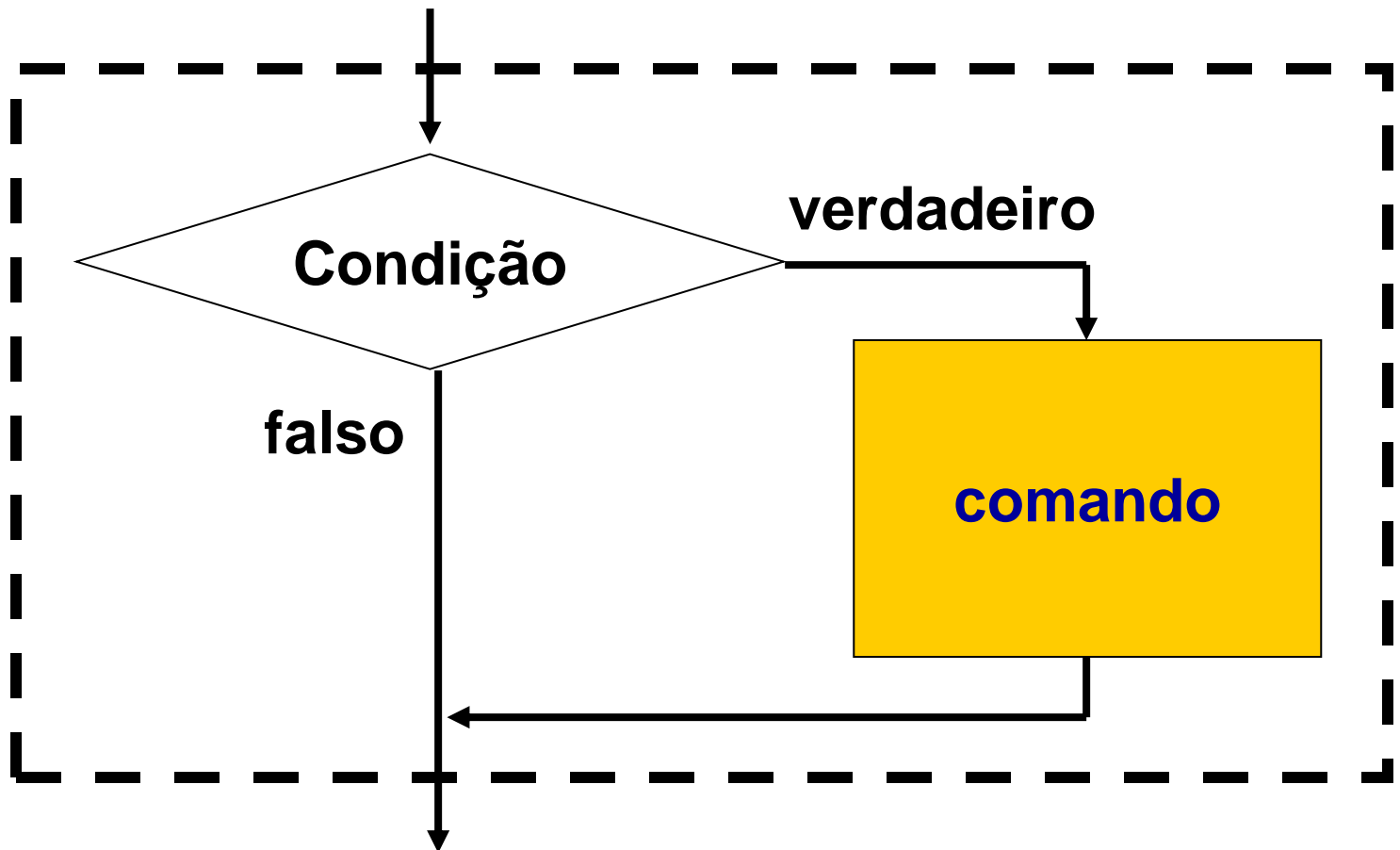
```
if (condição) {  
    comando1;  
    comando2;  
    comando3;  
}
```

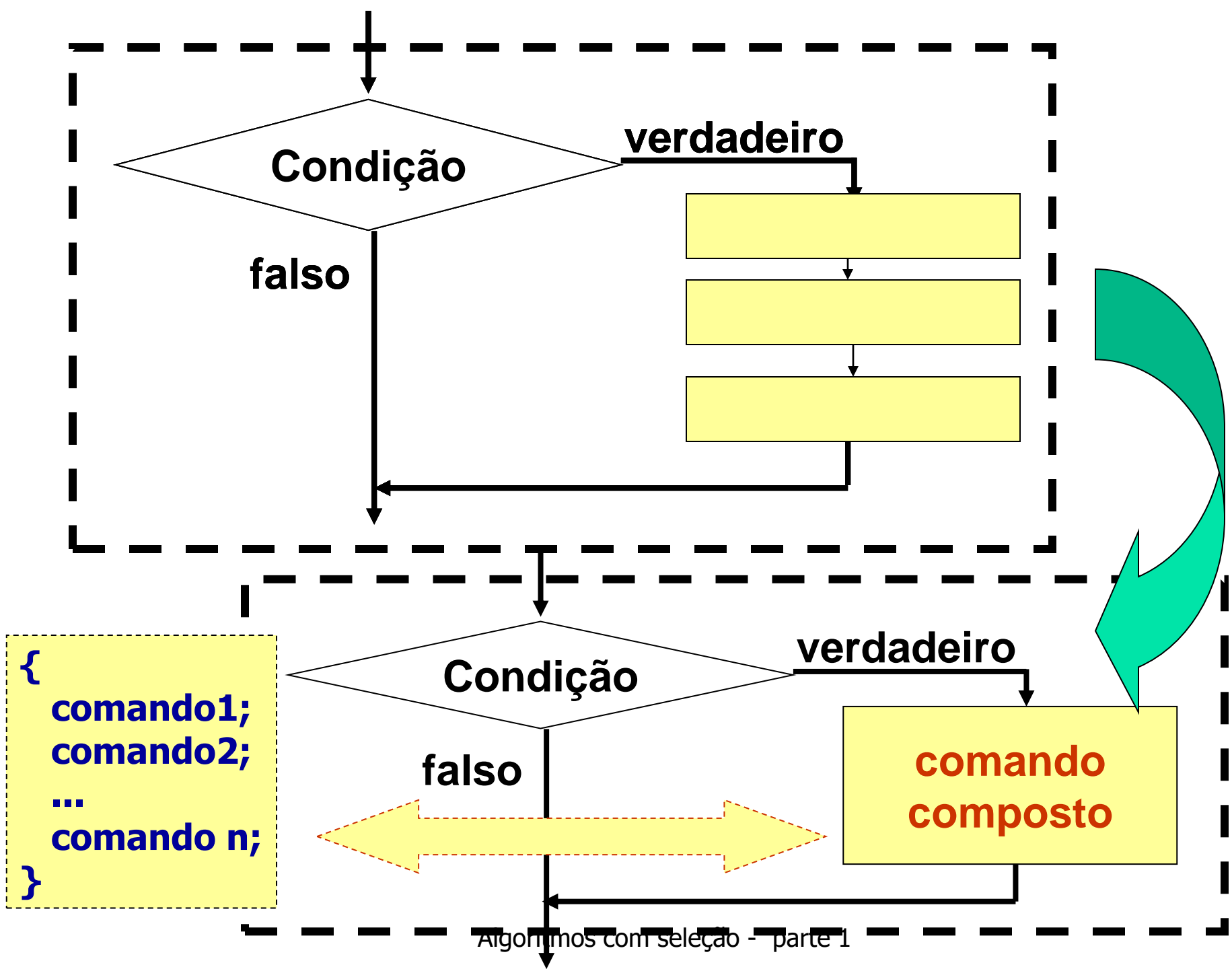
■ Funcionamento

- Se condição resultar verdadeira (diferente de zero), então o(s) comando(s) após o *if* será(ão) executado(s).



Seleção simples, 1 comando







Exemplos de seleção simples:

```
if (a < menor)
    menor = a;
```

```
if (a < menor) {
    menor = a;
    printf ("%d",
    menor);
}
```

- Observe que quando houver mais de um comando (ou *bloco*) subordinado a veracidade da condição, este bloco deve, obrigatoriamente, ser expresso entre chaves { ... }.



O comando *if... else...*

Estrutura de seleção dupla

■ Comando **if...else**

```
if (condição)
    comando;
else
    comando;
```

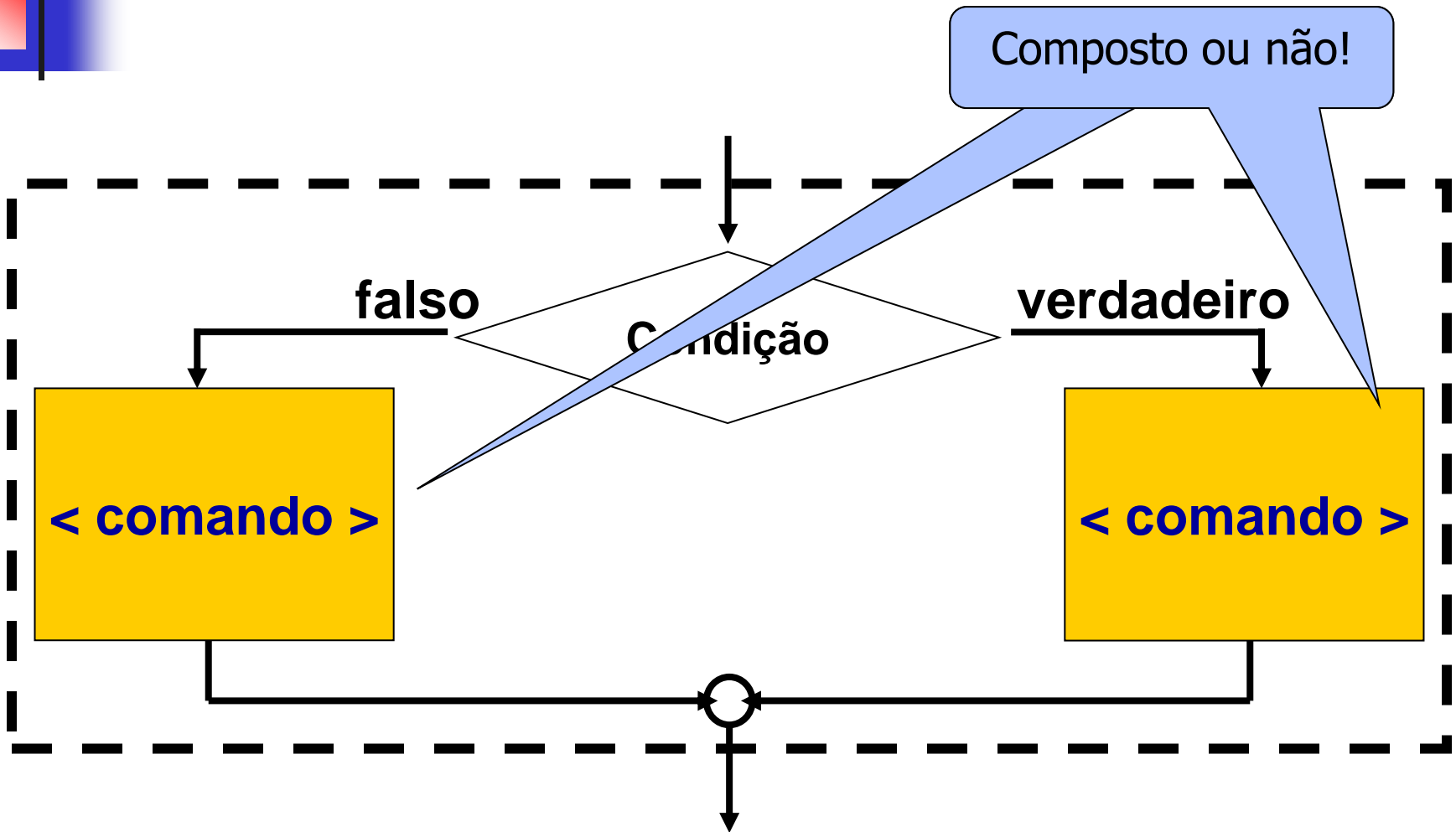
```
if (condição) {
    comando1;
    comando2;
} else {
    comando3;
    comando4;
}
```

■ Funcionamento

- se **condição** for **verdadeira** (diferente de zero) comando(s) após o *if* será(ão) executado(s);
- Caso contrário, se **condição** for **falsa** (igual a zero), comando(s) após o *else* será(ão) executado(s).

Observe que blocos subordinados devem aparecer entre chaves {...}

Estrutura de decisão dupla





Exemplo de seleção dupla

- **Em pseudo-código:**

Se (peso == peso_ideal)

Exibir "Vc está em forma!"

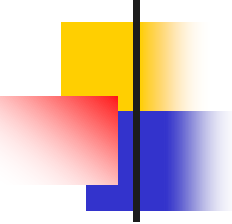
Senão

Exibir "Necessário fazer dieta!"

☛ Observe que na *seleção simples* somente é indicado a(s) ação(ões) a serem executadas quando a condição é *verdadeira*; caso contrário, essas ações são "puladas". Já a *seleção dupla* (if... else ...) permite ao programador especificar as diferentes ações que devem ser executadas quando a condição resultar *verdadeira* e quando a condição resultar *falsa*.

Retomando o Problema1:

Localização de um ponto no plano.



Dados um par de valores x e y , que representam as coordenadas de um ponto no plano, determinar a localização do ponto: se em um quadrante, em um dos eixos ou na origem.

- **Saída:** mensagem adequada
- **Entradas:** coordenadas x e y de um ponto
- **Processamento:**
série de testes, avaliando condições, para verificar em qual caso o par de valores enquadra-se.



```
// Inserir cabeçalho ...
```

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

```
#include <stdlib.h>
```

```
int main ( )
```

```
{
```

```
float x , y;
```

```
printf("\n Coordenadas: x = ");
```

```
scanf("%f", &x);
```

```
printf(" y = ");
```

```
scanf("%f", &y);
```

```
if ( x == 0 && y == 0 )
```

```
    printf("\n Ponto na origem");
```

```
if ( x > 0 && y > 0 )
```

```
    printf("\n Ponto no quadrante 1");
```

```
(...)
```

```
if ( x == 0 && y != 0 )
```

```
    printf("\n Ponto no eixo dos y");
```

```
if ( x != 0 && y == 0 )
```

```
    printf("\n Ponto no eixo dos x");
```

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

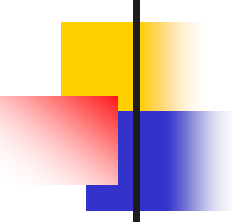
```
return 0;
```

```
}
```

Problema 1: "Localização de ponto no plano"

Se condição ① for verdadeira o comando ② após o *if* será executado.

Características da solução do Problema1:

- 
- Usa sucessivos *ifs* sem *else*.
 - Em cada condição avaliada é indicado apenas o que fazer se condição for verdadeira.
 - Se condição for falsa, o próximo comando será executado.
 - Como o programa usa apenas *seleção simples*, apesar de ele imprimir apenas uma mensagem de localização do ponto, **TODAS AS CONDIÇÕES SERÃO AVALIADAS!** Isto é **ineficiente!**
 - Você verá na Apresentação 5.2 como se resolve esse tipo de problema!
 - Dica: em determinadas situações, a solução converge para o uso de *ifs* encadeados (ou aninhados)!