

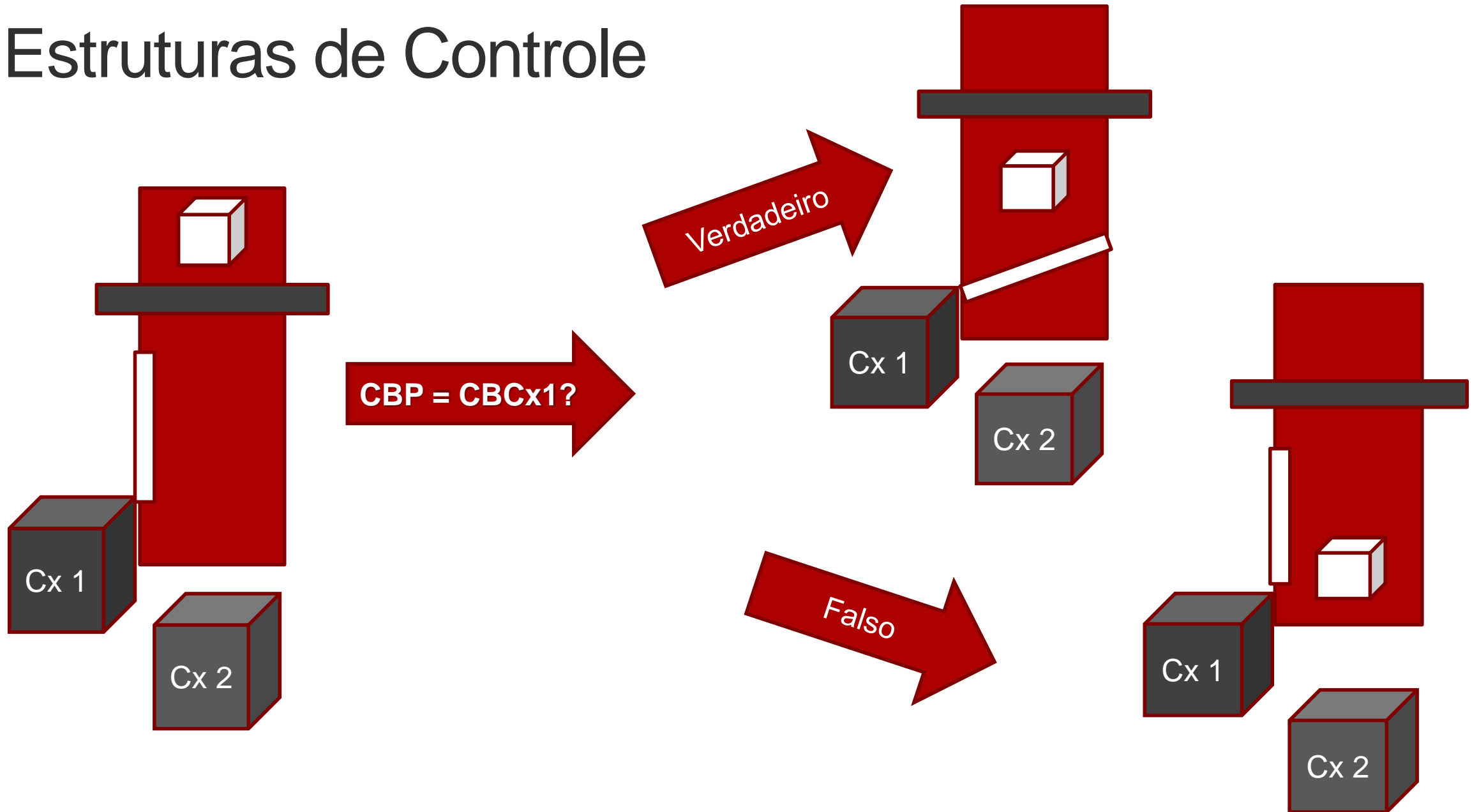
AULA DE ALGORITMO - 02

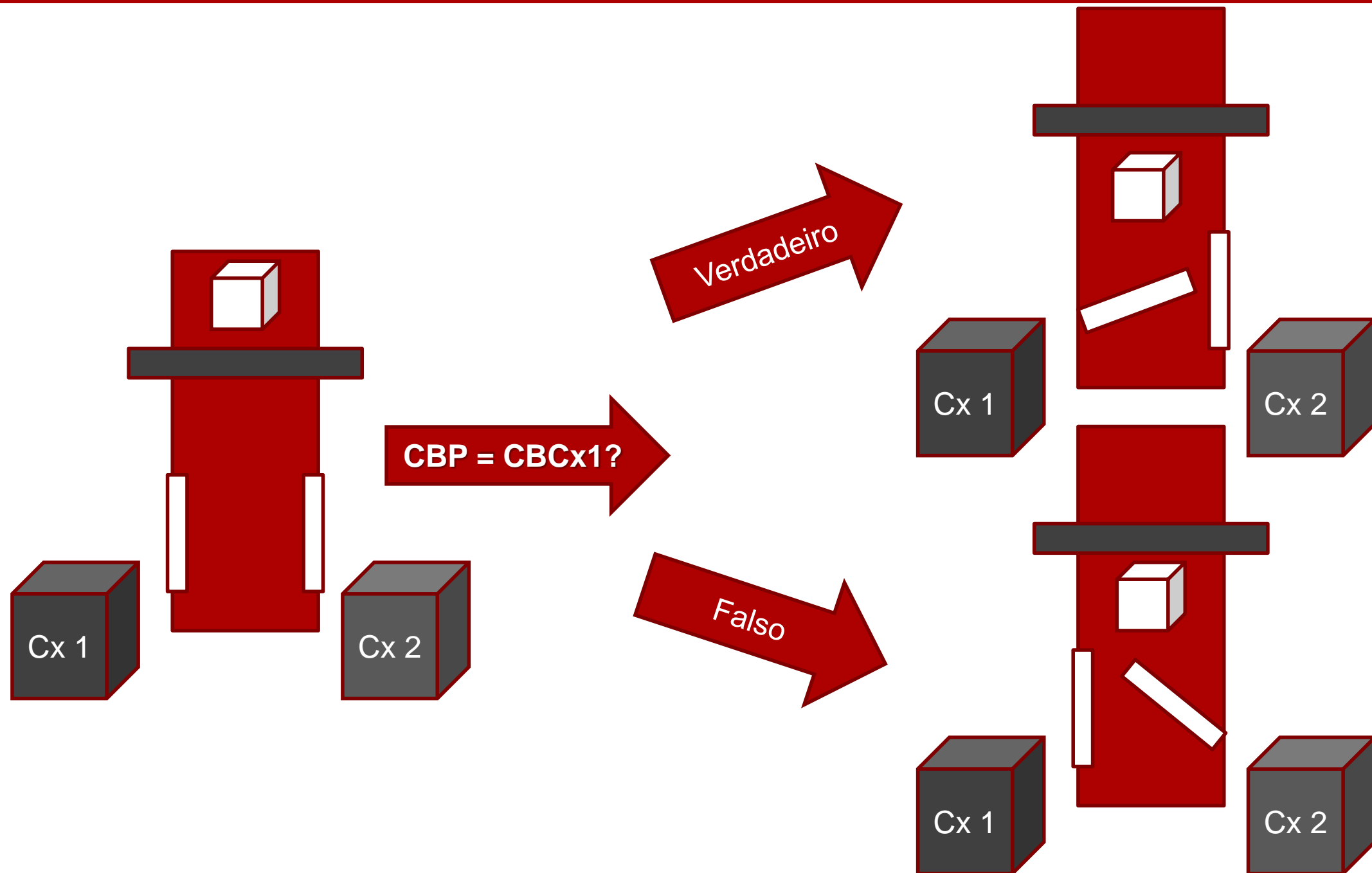
Profa. M. Sc. Valéria Maria Volpe

Estruturas de Controle

- Os algoritmos/programas desenvolvidos até o momento respeitam a execução sequencial dos comandos, que representam as ações lógicas que solucionam o problema que se quer informatizar.
- Porém, a estrutura sequencial não nos permite solucionar todos os problemas, pois muitas vezes, precisamos selecionar o que se quer executar, ou repetir a mesma ação para um conjunto de dados.
- Para solucionar esta dificuldade usamos as estruturas de controle:
 - De decisão (seleção)
 - De repetição

Estruturas de Controle





Estruturas de Controle

- **Estrutura de Controle de Decisão:**

- As estruturas de decisão (seleção) permitem escolher um grupo de ações (bloco de comandos) para ser executado quando determinada condição (comparação) for ou não satisfatória.
- As estruturas de decisão são:
 - Estrutura de decisão simples;
 - Estrutura de decisão composta;
 - Estrutura de decisão de múltipla escolha;
 - Estrutura de decisão encaixada.

Estruturas de Controle

- **Estrutura de Controle de Repetição:**

- As estruturas de repetição permitem executar um grupo de ações (bloco de comandos) uma ou mais vezes, criando assim uma estrutura de laço (loop). Esta repetição ocorrerá quando determinada condição (comparação) for satisfatório.
- As estruturas de repetição são:
 - Estrutura de repetição com teste no início;
 - Estrutura de repetição com teste no final;
 - Estrutura de repetição com variável de controle.

Estruturas de Decisão

- **Estrutura de Decisão Simples**

- A estrutura de decisão simples deve ser usada quando se tem uma ação (bloco de comando) que deve ser executada somente quando a condição for satisfatória.
- A estrutura de decisão simples testa a condição **antes** de executar a ação. Se a condição for satisfatória (verdadeira) as ações serão executadas.

Estruturas de Decisão

- **Sintaxe – Algoritmo**

se Condição então

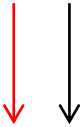
início **V**

→ Bloco de comandos que será executado somente se a

→ **condição** for satisfatória (**verdadeira**)

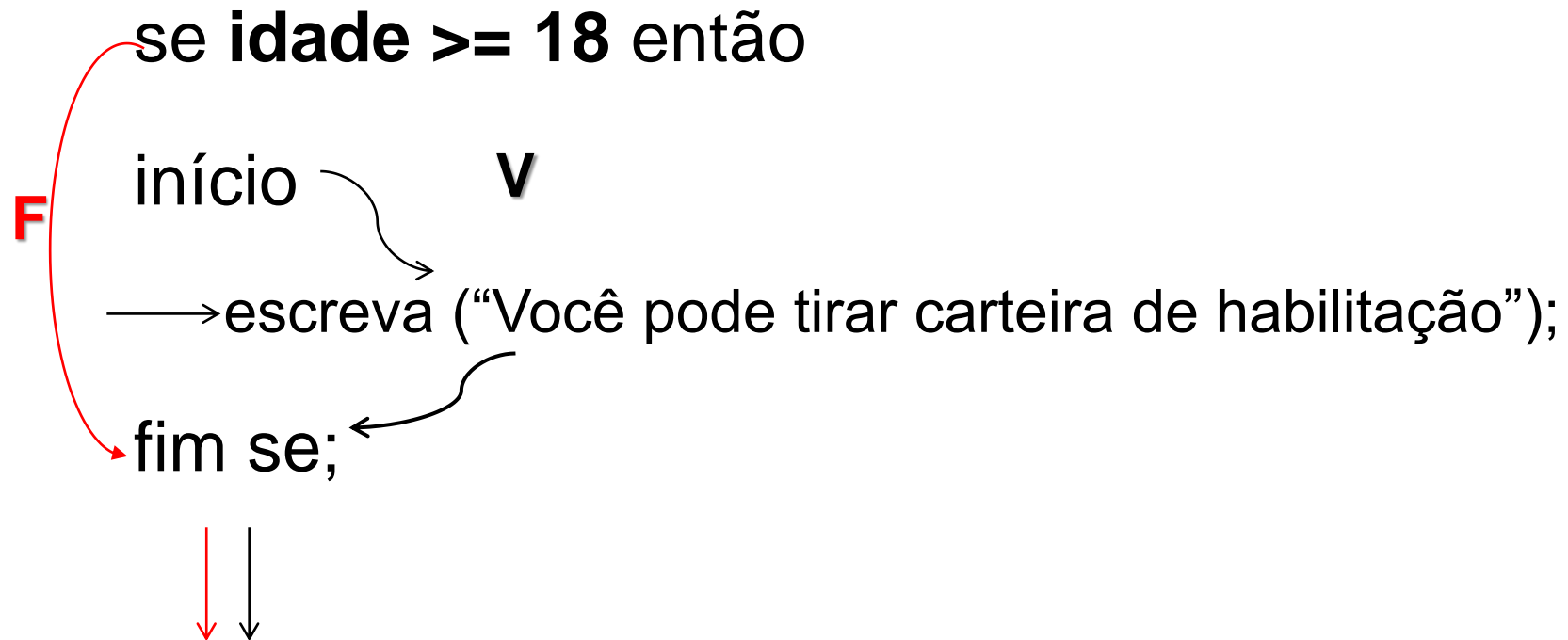
fim se; ↖

F



Estruturas de Decisão

- **Exemplo:**



Estruturas de Decisão

- Sintaxe – Linguagem C

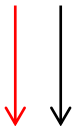
if (**Condição**)

{ **V**

—> Bloco de comandos que será executado somente se a **condição** for

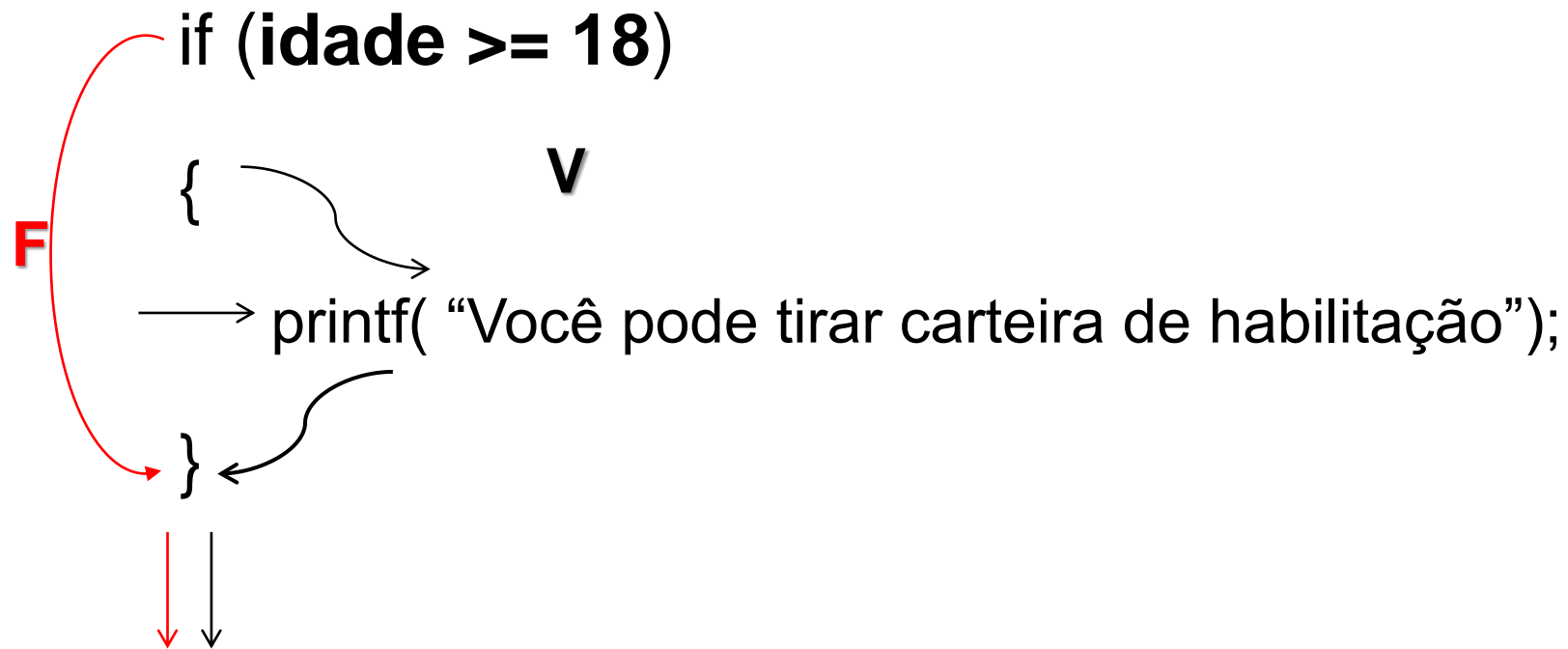
—> satisfatória (**verdadeira**)

}



Estruturas de Decisão

- Exemplo:



Estruturas de Decisão

- **Exercícios:**

1) Faça um algoritmo/programa para ler um n° inteiro e verificar se o n° lido é múltiplo de 3. Mostre a mensagem: “Precisamos de CHUVA urgentemente!”

- **Ações:**

- Ler um número
- Verificar se o número é múltiplo de 3
- Mostrar a mensagem

- **Variável**

- Num
- Resto

Estruturas de Decisão

Algoritmo Múltiplo de 3;

Início

var

inteiro: num, resto;

leia (num);

resto \leftarrow num mod 3;

se resto = 0 então

início

escreva(num, "É múltiplo de 3");

fim se;

escreva ("Precisamos de CHUVA urgentemente!");




Fim.

Estruturas de Decisão



```
#include<stdio.h>
#include<stdlib.h>
main ()
{
    int num, resto;
    printf("\n Digite um nº inteiro: ");
    scanf("%d", &num);
    resto = num % 3;
    if (resto == 0)
    {
        printf("\n %d é múltiplo de 3", num);
    }
    printf("\n\n Precisamos de CHUVA urgentemente\n\n");
    system ("pause");
}
```

Estruturas de Decisão

• Exercícios

- 2) Faça um algoritmo para ler a idade de uma pessoa e verificar se ela pode votar nas próximas eleições. 
- 3) Faça um algoritmo para ler as notas das duas provas e a média de trabalho de um aluno, calcular sua média final e verificar se ele está aprovado. 
- 4) Faça um algoritmo/programa para ler o valor total de uma compra e o código do pagamento. Se o código do pagamento for 1 dar 5% de desconto. Mostre o valor a ser pago. 

Exercícios

- 5) Faça um algoritmo para ler um número inteiro e verificar se o número lido é ímpar. 
- 6) Faça um algoritmo que leia dois números, calcule e mostra a divisão do primeiro número digitado pelo segundo. Lembre-se que não existe divisão por zero.
- 7) Faça um algoritmo para ler o salário do funcionário e a quantidade de anos que ele trabalha na empresa. Se fizer mais de 10 anos que ele trabalha na empresa, dar aumento de 10% de salário. 

EXERCÍCIO PARA ENTREGAR

- 1) Faça um algoritmo/programa em Ling. C para ler o salário do funcionário e a idade do seu filho. Para filho com menos de 14 anos acrescentar R\$ 350,00 no salário do funcionário. Mostre o salário do funcionário.

Solução Exercício 2

Algoritmo Votação;

Início

var

inteiro: idade;

leia (idade);

se idade \geq 16 então

início

escreva("Você pode votar nas próximas eleições");

fim se;

escreva ("Sua idade é: ", idade, " anos"); // opcional

Fim.

| idade | idade \geq 16 | Saída de dados |
|-------|-----------------------|--|
| 16 | 16 \geq 16 V | Você pode votar... Sua idade é: 16 anos |
| idade | idade \geq 16 | Saída de dados |
| 34 | 34 \geq 16 V | Você pode votar... Sua idade é: 34 anos |
| idade | idade \geq 16 | Saída de dados |
| 14 | 14 \geq 16 F | Sua idade é: 14 anos |



Solução Exercício 3

Algoritmo Média Final;

Início

var

real: notaProva1, notaProva2, mediaProva, mediaTrab, mediaFinal;

leia (notaProva1);

leia (notaProva2);

leia (mediaTrab);

$mediaProva \leftarrow (notaProva1 + notaProva2) / 2.0;$

$mediaFinal \leftarrow 0.7 * mediaProva + 0.3 * mediaTrab;$

se mediaFinal \geq 6.0 então

início

escreva("Aluno Aprovado");

fim se;

escreva ("Média Final : ", mediaFinal);

Fim.



Solução 1 Exercício 4

Algoritmo Valor da Compra;

Início

var

inteiro: codigoPagto;

real: valorCompra;

leia (valorCompra);

leia (codigoPagto);

se codigoPagto = 1 então

início

valorCompra \leftarrow valorCompra – (valorCompra * 0.05);

fim se;

escreva (“Valor Compra R\$ ”, valorCompra);

Fim.



Solução 2 Exercício 4

Algoritmo Valor da Compra;

Início

var

inteiro: codigoPagto;

real: valorCompra, pagamento, desconto;

leia (valorCompra);

leia (codigoPagto);

desconto \leftarrow 0.0;

se codigoPagto = 1 então

início

desconto \leftarrow valorCompra * 0.05;

fim se;

pagamento \leftarrow valorCompra - desconto;

escreva ("Valor Compra R\$ ", valorCompra, " Desconto R\$", desconto, " Total R\$ ", pagamento);

Fim.



Exercício 5

Algoritmo Número Ímpar;

Início

var

inteiro: num;

leia (num);

se (num mod 2) <> 0 então

início

escreva(num, “ é ímpar”);

fim se;

Fim.



Exercício 7

Algoritmo Aumento Salario;

Início

var

inteiro: tempoServico;

real: salario, salarioFinal, aumento;

leia (tempoServico);

leia (salario);

aumento \leftarrow 0.0;

se tempoServico > 10 então

início

aumento \leftarrow salario* 0.10;

fim se;

salarioFinal \leftarrow salario + aumento;

escreva (“Salário R\$ ”, salario, “ Aumento R\$ ”, aumento, “ Salário final R\$ “, salarioFinal);

Fim.



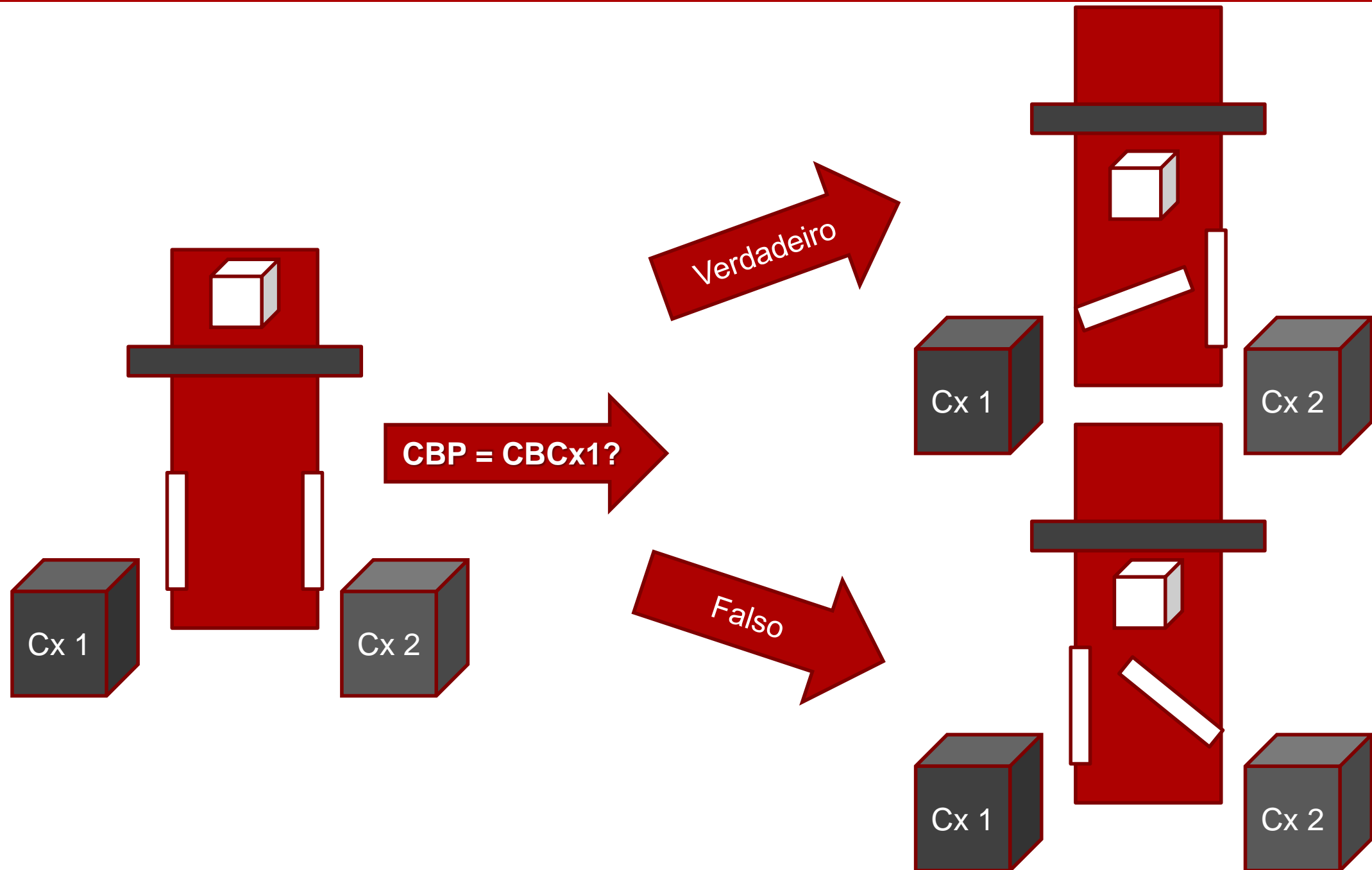
AULA DE ALGORITMO ESTRUTURA DE DECISÃO COMPOSTA

Profa. M. Sc. Valéria Maria Volpe

Estruturas de Decisão

- **Estrutura de Decisão Composta**


- Esta estrutura deve ser usada quando se tem duas alternativas dependendo de **uma mesma condição**, sendo que uma alternativa será para quando a condição for satisfatória (verdadeira) e outra para quando a condição não for satisfatória (falsa).




Estruturas de Decisão

- Sintaxe – Algoritmo

se condição então

início  **V**

→ Bloco de comandos que será executado quando a **condição** for satisfatória (verdade);

fim 

senão

F início 

→ Bloco de comandos que será executado quando a **condição** não for satisfatória (falsa);

fim se; 



Estruturas de Decisão

- **Exemplo:**

se idade \geq 18 então

início \downarrow

→ escreva ("Você pode tirar carteira de habilitação");

fim ←

senão

F **início** →

→ escreva("Você não pode tirar carteira de habilitação");

fim se; ←



Estruturas de Decisão

- Sintaxe – Linguagem C

if (condição)

{ **V**

Bloco de comando que será executado quando a **condição** for satisfeita (verdadeira);

}

F **else**

{

Bloco de comando que será executado quando a **condição** não for satisfeita (falsa);

}



Estruturas de Decisão

- **Exemplo:**

```
if (idade >= 18)
{
    printf("Você pode tirar carteira de habilitação");
}
else
{
    printf("Você não pode tirar carteira de habilitação");
}
```

The diagram illustrates the execution flow of the code. A red arrow labeled 'F' (False) points from the condition `idade >= 18` to the `else` block. A black arrow labeled 'V' (Verdadeiro/True) points from the condition to the first `{` block. Inside the `if` block, a black arrow points to the `printf` statement. Inside the `else` block, a red arrow points to the `printf` statement. At the bottom, two vertical arrows (one red, one black) point downwards from the end of the `if-else` structure, indicating the flow continues after the decision.

Estruturas de Decisão

- **Exemplos**
- Faça um algoritmo/programa para ler o ano de nascimento de uma pessoa e o ano atual. Calcule a idade dessa pessoa e verifique se ela pode ou não votar.
 - Ler ano nascimento
 - Ler ano atual
 - Calcular idade
 - Verificar se pode ou não votar
- **Variáveis**
 - anoNasc
 - AnoAtual
 - Idade

Estruturas de Decisão

Algoritmo pode ou não votar;

Início

var

inteiro: anoNasc, anoAtual, idade;

leia (anoNasc);

leia (anoAtual);

idade \leftarrow anoAtual – anoNasc;

escreva (“Idade: “, idade);

se idade \geq 16 então

início

escreva(“Você pode votar”);

fim

senão

início

escreva(“Você não pode votar”);

fim se;

Fim.

Estruturas de Decisão

```
#include<stdio.h>
#include<stdlib.h>
main()
{
    int anoNasc, anoAtual, idade;
    printf("\n Digite o seu ano de nascimento: ");
    scanf("%d", & anoNasc);
    printf("\n Digite ano atual: ");
    scanf("%d", & anoAtual);
    idade = anoAtual – anoNasc;
    printf("Idade: %d", idade);
    if (idade >= 16 )
    {
        printf("\n Você pode votar!! \n");
    }
    else
    {
        printf("\n Você não pode votar \n");
    }
    system ("pause");
}
```

Estruturas de Decisão

- Faça um algoritmo/programa para ler a altura e o sexo de uma pessoa e calcular seu peso ideal, sabendo que a fórmula para calcular o peso ideal é:
 - Masculino: $(72.7 * altura) - 58.0$
 - Feminino: $(62.1 * altura) - 44.7$
- Ações:
 - Ler altura e sexo
 - Verificar o sexo M (Masculino) ou F (Feminino)
 - Mostrar o resultado
- Variáveis:
 - Altura
 - Sexo
 - PesoIdeal

Estruturas de Decisão

Algoritmo Peso Ideal;

Início

var

real: altura, pesoldeal;

caracter: sexo;

leia (altura);

leia (sexo);

se sexo = 'f' ou sexo = 'F' então

início

pesoldeal \leftarrow (62.1 * altura) - 44.7;

fim

senão

início

pesoldeal \leftarrow (72.7 * altura) - 58.0;

fim se;

escreva ("Seu peso ideal é: ", pesoldeal);

Fim.

Estruturas de Decisão

```
#include<stdio.h>
#include<stdlib.h>
main()
{
    float altura, pesoldeal;
    char sexo;
    printf("\n Digite a sua altura: ");
    scanf("%f", & altura);
    printf("\n Digite seu sexo: ");
    fflush(stdin); //serve para limpar dados do padrão de entrada (teclado)
    scanf("%c", & sexo);
    if(sexo == 'f' || sexo == 'F' )
    {
        pesoldeal = (62.1 * altura) - 44.7;
    }
    else
    {
        pesoldeal = (72.7 * altura) - 58.0;
    }
    printf("\n Seu peso Ideal é: %.2f Kg", pesoldeal);
    system ("pause");
}
```

Estruturas de Decisão

- **Exercícios:**

1) Faça um algoritmo/programa que irá ler dois números inteiros e mostrá-los em ordem crescente.




2) Uma empresa deseja aumentar o preço de seus produtos de acordo com o departamento que o produto pertence. Se o produto for do departamento 1 o aumento será de 5.0%, caso contrário será de 7.4%. Faça um algoritmo/programa para calcular e mostrar o preço do produto com o aumento dado.



Estrutura de Decisão

- Exercícios

- 3) Faça um algoritmo/programa para ler um caracter e dizer se é uma letra ou não.
 - 4) Faça um algoritmo para ler as notas das duas provas e a média de trabalho de um aluno, calcular sua média final e verificar se ele está aprovado ou não.
- 

Solução Exercício 1

Algoritmo Ordem Crescente;

Início

var

inteiro: num1, num2;

leia (num1);

leia (num2);

se num1 < num2 então

início

escreva("Ordem Crescente: ", num1, num2);

fim

senão

início

escreva("Ordem Crescente: ", num2, num1);

fim se;

Fim.



Solução 1 - Exercício 2

Algoritmo Aumento Preço;

Início

var

inteiro: codigoDepto;

real: precoProduto, precoNovo;

leia (precoProduto);

leia (codigoDepto);

se codigoDepto = 1 então

início

precoNovo \leftarrow precoProduto + precoProduto * 0.05;

fim

senão

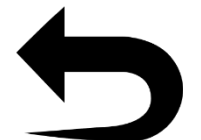
início

precoNovo \leftarrow precoProduto + precoProduto * 0.074;

fim se;

escreva("Preço Novo: ", precoNovo);

Fim.



Solução 2 – Exercício 2

Algoritmo Aumento Preço;

Início

var

inteiro: codigoDepto;

real: precoProduto, precoNovo, aumento;

leia (precoProduto);

leia (codigoDepto);

se codigoDepto = 1 então

início

aumento \leftarrow precoProduto * 0.05;

fim

senão

início

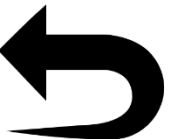
aumento \leftarrow precoProduto * 0.074;

fim se;

precoNovo \leftarrow precoProduto + aumento;

escreva("Preço do Produto R\$", precoProduto, "Aumento R\$", aumento, "Preço Novo R\$ ", precoNovo);

Fim.



Solução Exercício 3

Algoritmo Letra ou Não;

Início

var

character: carac;

leia (carac);

se (carac >= 'A' E carac <= 'Z') OU (carac >= 'a' E carac <= 'z') então

início

escreva("É letra:");

fim

senão

início

escreva("Não é letra ");

fim se;

Fim.

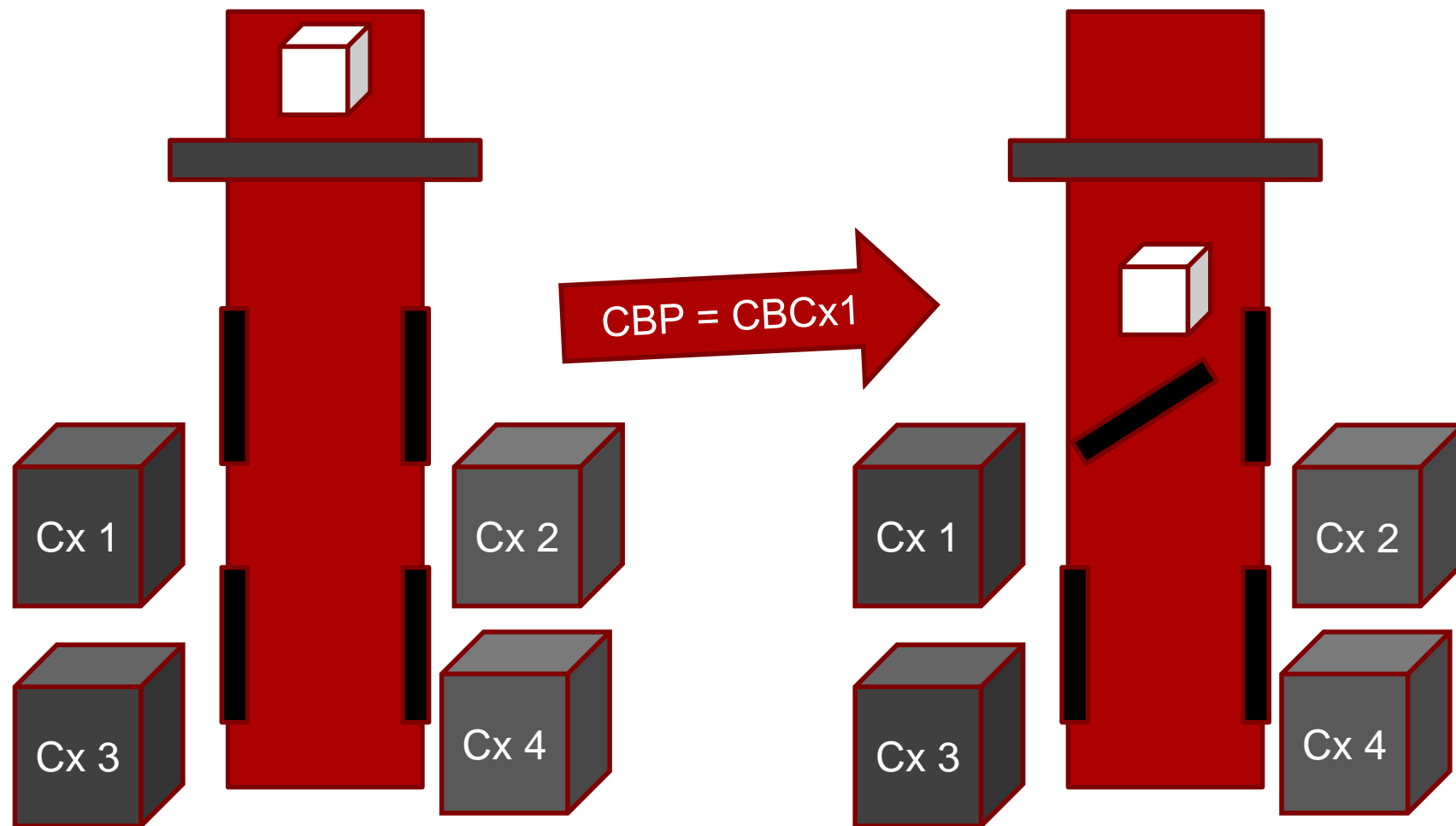


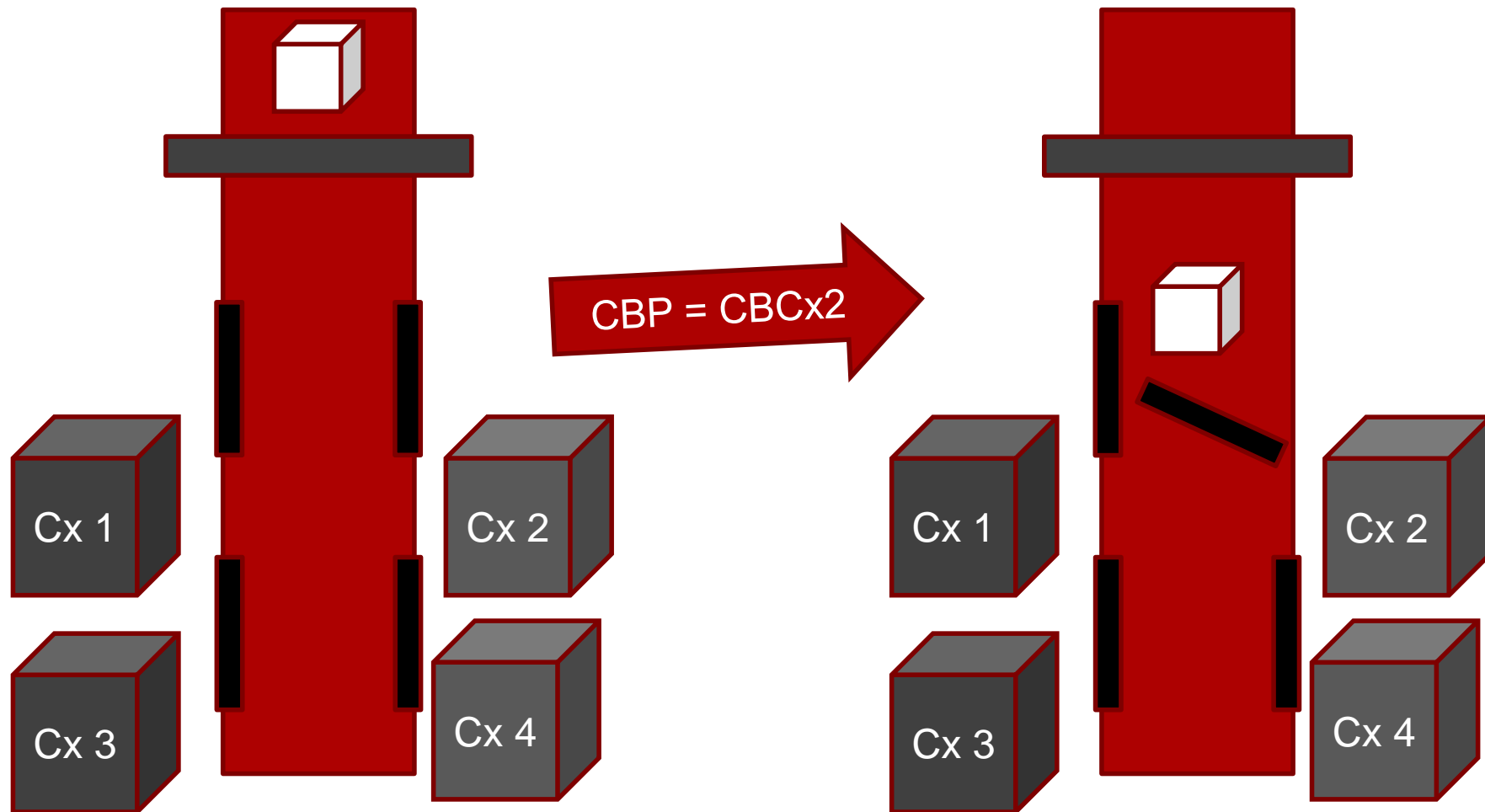
AULA DE ALGORITMO ESTRUTURA DE DECISÃO DE MÚLTIPLA ESCOLHA E ENCAIXADA PARTE 02

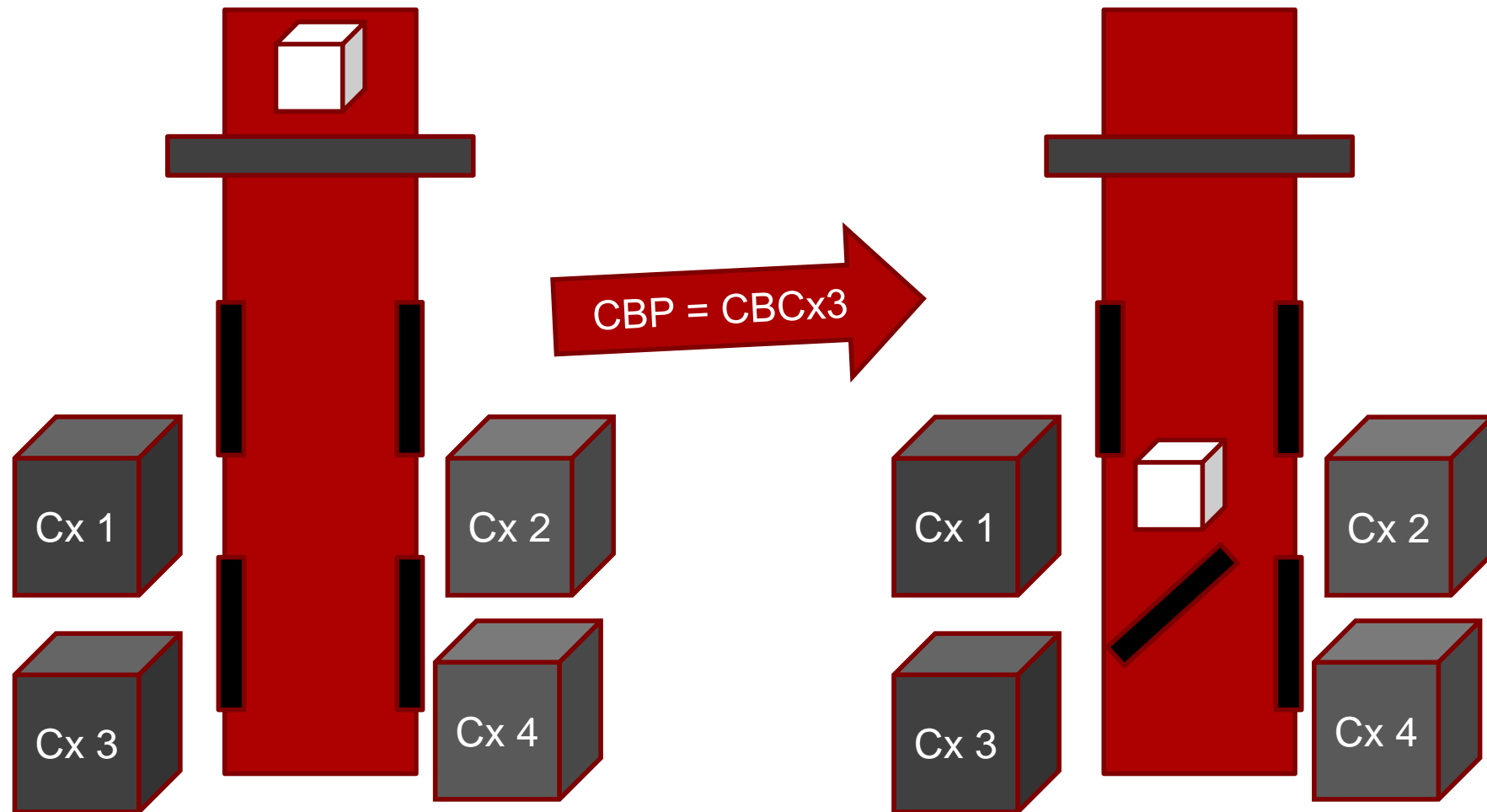
Profa. M. Sc. Valéria Maria Volpe

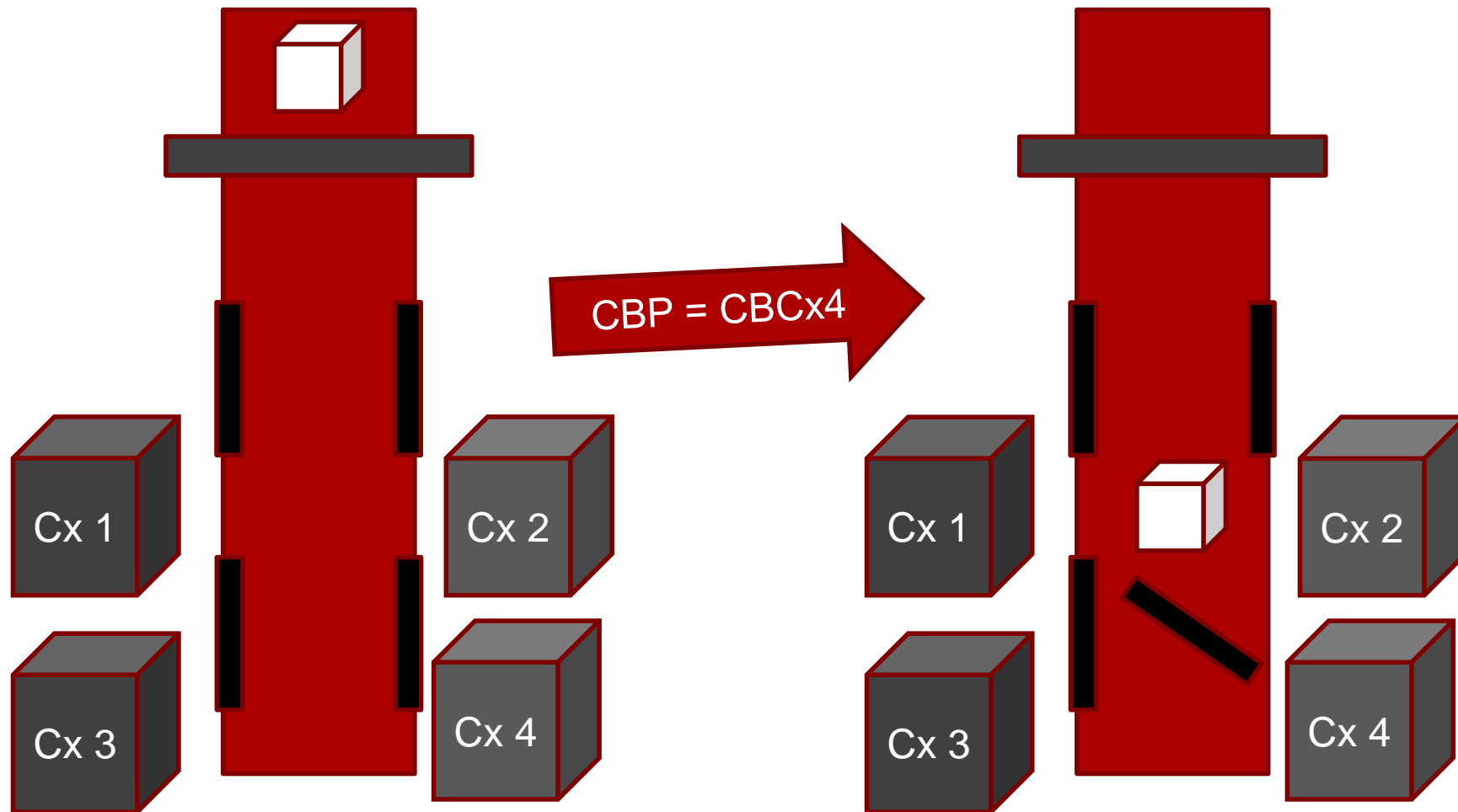
AULA DE ALGORITMO ESTRUTURA DE DECISÃO DE MÚLTIPLA ESCOLHA

Profa. M. Sc. Valéria Maria Volpe









Estruturas de Decisão

- **Estrutura de decisão de múltipla escolha**
- Esta estrutura pode ser usada quando um conjunto de valores discretos precisa ser testado e um conjunto de ações (bloco de comandos) diferentes são associadas a esses valores.

Estruturas de Decisão

- Sintaxe – Algoritmo

escolha (**nomeVar**)

caso v1: Bloco de comandos que será executado quando **NomeVar = v1**

caso v2: Bloco de comandos que será executado quando **NomeVar = v2**

.

.

.

caso vn: Bloco de comandos que será executado quando **NomeVar = vn**

caso contrário: Bloco de comandos que será executado quando **NomeVar \neq v1, v2, ..., vn**

fim escolha;

Estruturas de Decisão

- **Esta estrutura só pode ser usada quando:**
 - A comparação for uma comparação de igualdade.
 - Quando a variável (NomeVar) usada para fazer a comparação for N° Inteiro ou um único caracter.

Estruturas de Decisão

- **Exemplo**

escolha (tipoPagto)

caso 1: escreva(“compra a vista com desconto”);

caso 2: escreva(“compra a vista no cartão de crédito”);

caso 3: escreva(“compra a prazo no boleto”);

caso 4: escreva(“compra a prazo no cartão de crédito”);

caso contrário: escreva(“Opção inválida”);

fim escolha;

Estruturas de Decisão

- **Exemplo**

switch (tipoPagto)

{

case 1: printf("\n Compra a vista com desconto \n");

break;

case 2: printf("\n Compra a vista no cartão de crédito \n");

break;

case 3: printf("\n Compra a prazo no boleto \n");

break;

case 4: printf("\n Compra a prazo no cartão de crédito \n");

break;

default : printf("\n Opção inválida \n");

}

Estruturas de Decisão

- **Exemplo**

escolha (letra)

caso 'A', 'a': escreva ("A próxima vogal é E");

caso 'E', 'e': escreva ("A próxima vogal é I");

caso 'I', 'i': escreva ("A próxima vogal é O");

caso 'O', 'o': escreva ("A próxima vogal é U");

caso 'U', 'u': escreva ("Esta é a última vogal");

caso contrário: escreva ("Você não digitou uma vogal");

fim escolha;

Estruturas de Decisão

- Exemplo

```
switch (letra)
{
    case 'A': case 'a': printf("\n A próxima vogal é E \n");
    break;
    case 'E': case 'e': printf("\n A próxima vogal é I \n ");
    break;
    case 'I': case 'i': printf("\n A próxima vogal é O \n");
    break;
    case 'O': case 'o': printf("\n A próxima vogal é U \n");
    break;
    case 'U': case 'u': printf("\n Esta é a última vogal \n");
    break;
    default: printf("\n Você não digitou uma vogal \n");
}
```


Estruturas de Decisão

- **Exercícios**
- Faça um algoritmo para ler um número inteiro e mostre o nome do mês representado pelo valor numérico lido.
- Ações
 - - Ler número
 - - Verificar valor digitado
 - - Mostrar o nome do mês
- Variável
 - Num – inteiro

Estruturas de Decisão

Algoritmo Mês;

Início

var

inteiro: num;

leia (num);

escolha (num)

caso 1: escreva ("Janeiro");

caso 2: escreva ("Fevereiro");

caso 3: escreva ("Março");

caso 4: escreva ("Abril");

caso 5: escreva ("Maio");

caso 6: escreva ("Junho");

caso 7: escreva ("Julho");

caso 8: escreva ("Agosto");

caso 9: escreva ("Setembro");

caso 10: escreva ("Outubro");

caso 11: escreva ("Novembro");

caso 12: escreva ("Dezembro");

caso contrário: escreva ("O valor digitado não corresponde a um mês");

fim escolha;

Fim.

Estruturas de Decisão

```
#include<stdio.h>
main()
{
    int num;
    printf("\n Digite um numero inteiro: ");
    scanf("%d", &num);
    switch (num)
    {
        case 1 : printf( "\n Janeiro \n ");      break;
        case 2 : printf("\n Fevereiro \n");      break;
        case 3 : printf("\n Março \n");          break;
        case 4 : printf("\n Abril \n");           break;
        case 5 : printf( "\n Maio \n");           break;
        case 6 : printf( "\n Junho \n");          break;
        case 7 : printf( "\n Julho \n");          break;
        case 8 : printf( "\n Agosto \n");         break;
        case 9 : printf( "\n Setembro \n");       break;
        case 10 : printf( "\n Outubro \n");       break;
        case 11 : printf( "\n Novembro \n");      break;
        case 12 : printf( "\n Dezembro \n");      break;
        default: printf( "\n O valor digitado não corresponde a um mês \n");
    } //Fim do switch
}
```

Estruturas de Decisão

- Exercícios:
 - 1) Faça um algoritmo/programa para ler um número inteiro que corresponde a um mês. De acordo com o valor digitado mostre a quantidade de dias que o mês tem. Desconsidere ano bissexto.
 - 2) Faça um algoritmo/programa que leia dois números reais e calcule as operações de acordo com a escolha do usuário, pelo código da tabela abaixo:

| Escolha do usuário | Operação realizada |
|--------------------|--|
| 1 | Média Aritmética |
| 2 | Diferença do maior pelo menor |
| 3 | Multiplicação entre os números |
| 4 | Divisão do primeiro número pelo segundo digitado |



Estruturas de Decisão

- 3) Faça um algoritmo para ler dois números inteiros e um caractere. O caractere digitado irá representar uma operação aritmética (+, -, *, /, %) . De acordo com o operador digitado realize a operação desejada e mostre os resultados.
- 4) Faça um programa que receba o preço de um produto e o seu código de origem e mostre na tela a sua procedência e o preço do produto. A procedência obedece a tabela abaixo:



| Código de origem | Procedência |
|------------------|---------------------|
| 1 | Região Sul |
| 2 | Região Norte |
| 3 | Região Centro-Oeste |
| 4 | Região Nordeste |
| 5 ou 6 | Região Sudeste |
| 7, 8 ou 9 | Região Centro-Oeste |

Estruturas de Decisão Exercício para Entregar

- Uma confeitaria está vendendo doces com a seguinte tabela de preços.

| Tipo da doce | Até 5 Kg | Acima de 5 Kg |
|--------------|------------------|------------------|
| 1 - Trufa | R\$ 25,00 por Kg | R\$ 22,00 por Kg |
| 2 - Torta | R\$ 24,45 por Kg | R\$ 22,25 por Kg |
| 3 – Bolo | R\$ 30,50 por Kg | R\$ 28,90 por Kg |

- Escreva um **ALGORITMO** para ler o tipo e a quantidade (em Kg) de doce. Calcule e mostre o valor a ser pago pelo cliente.

Estruturas de Decisão

Algoritmo Dia Mês;

Início

var

inteiro: num;

leia (num);

escolha (num)

caso 1, 3, 5, 7, 8, 10, 12: escreva("31 dias");

caso 2: escreva ("28 dias");

caso 4, 6, 9, 11: escreva ("30 dias");

caso contrário: escreva ("O valor digitado não corresponde a um mês");

fim escolha;

Fim.

Estruturas de Decisão

```
#include<stdio.h>
main()
{
    int num;
    printf("\n Digite um numero inteiro: ");
    scanf("%d", &num);
    switch (num)
    {
        case 1 :
        case 3 :
        case 5:
        case 7:
        case 8:
        case 10:
        case 12: printf("\n 31 dias \n");           break;
        case 2 : printf("\n 28 dias \n");           break;
        case 4:
        case 6 :
        case 9 :
        case 11 : printf( "\n 30 dias \n");           break;
        default: printf( "\n O valor digitado não corresponde a um mês \n");
    } //Fim do switch
}
```


Estruturas de Decisão

Algoritmo Operações;

Início

var

real: x, y, resultado;

inteiro: opcao;

leia(x);

leia(y);

leia(opcao);

escolha (opcao)

caso 1: resultado \leftarrow (x + y) / 2.0;

escreva("Média aritmética = ", resultado);

caso 2: se x \geq y então

início

resultado \leftarrow x - y;

fim

senão

início

resultado \leftarrow y - x;

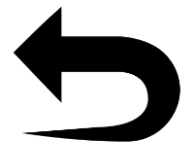
fim se;

escreva("Subtração = ", resultado);

// Continua no próximo slide

Estruturas de Decisão

```
caso 3: resultado  $\leftarrow$  x * y;  
        escreva("Multiplicação = ", resultado);  
caso 4: se y  $\neq$  0.0 então  
        início  
            resultado  $\leftarrow$  x / y;  
            escreva("Divisão = ", resultado);  
        fim  
        senão  
        início  
            escreva(" Não existe divisão por zero");  
        fim se;  
caso contrário: escreva("Operação inválida");  
fim escolha;  
Fim.
```



Estruturas de Decisão

Algoritmo Calculadora;

Inicio

var

inteiro: x, y, resultado;

caracter: operador;

leia(x);

leia(y);

leia(operador);

escolha (operador)

caso '+': resultado \leftarrow x + y;

escreva("Soma = ", resultado);

caso '-': resultado \leftarrow x - y;

escreva("Subtração = ", resultado);

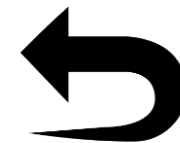
caso '*': resultado \leftarrow x * y;

escreva("Multiplicação = ", resultado);

// Continua no próximo slide

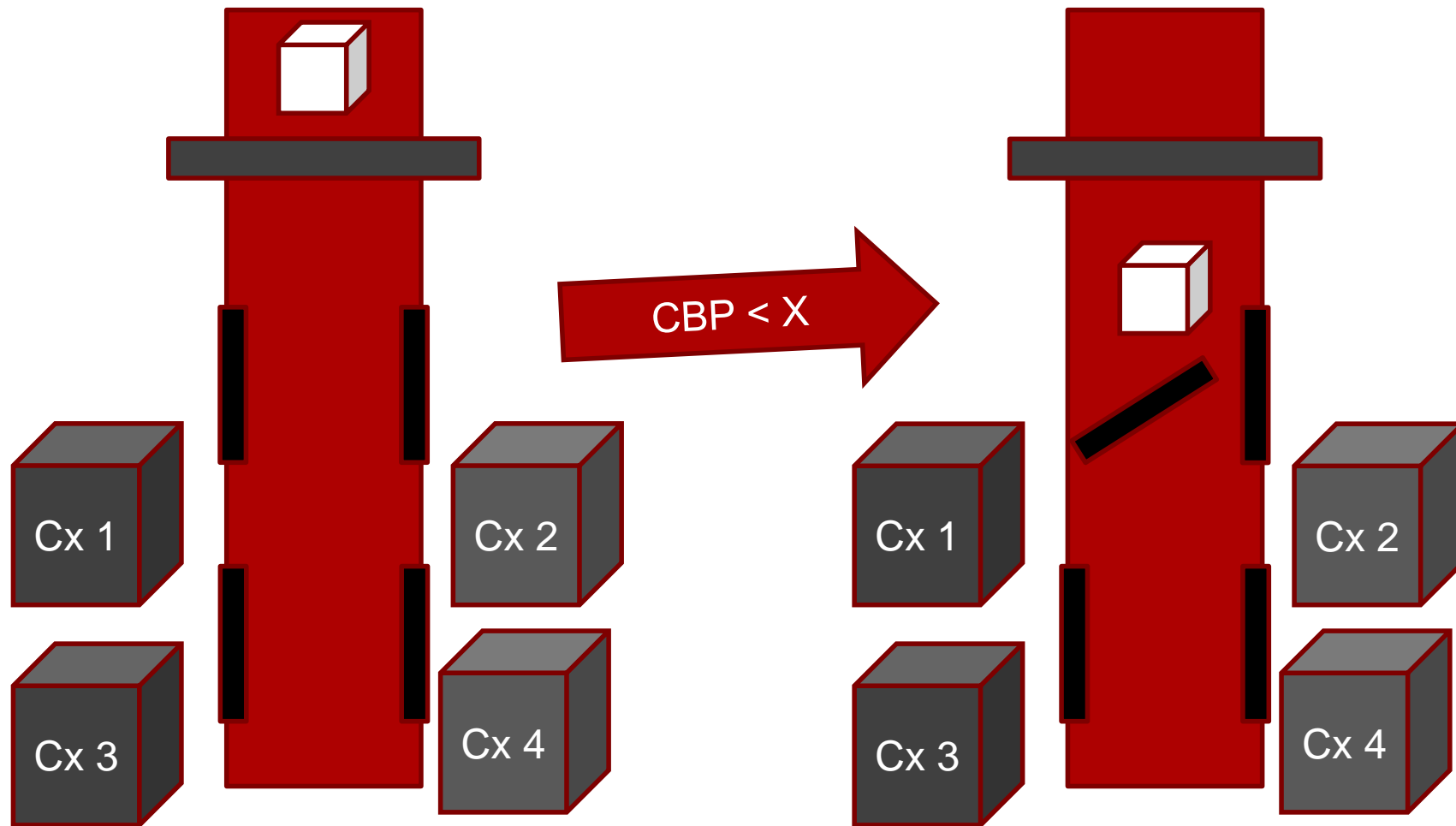
Estruturas de Decisão

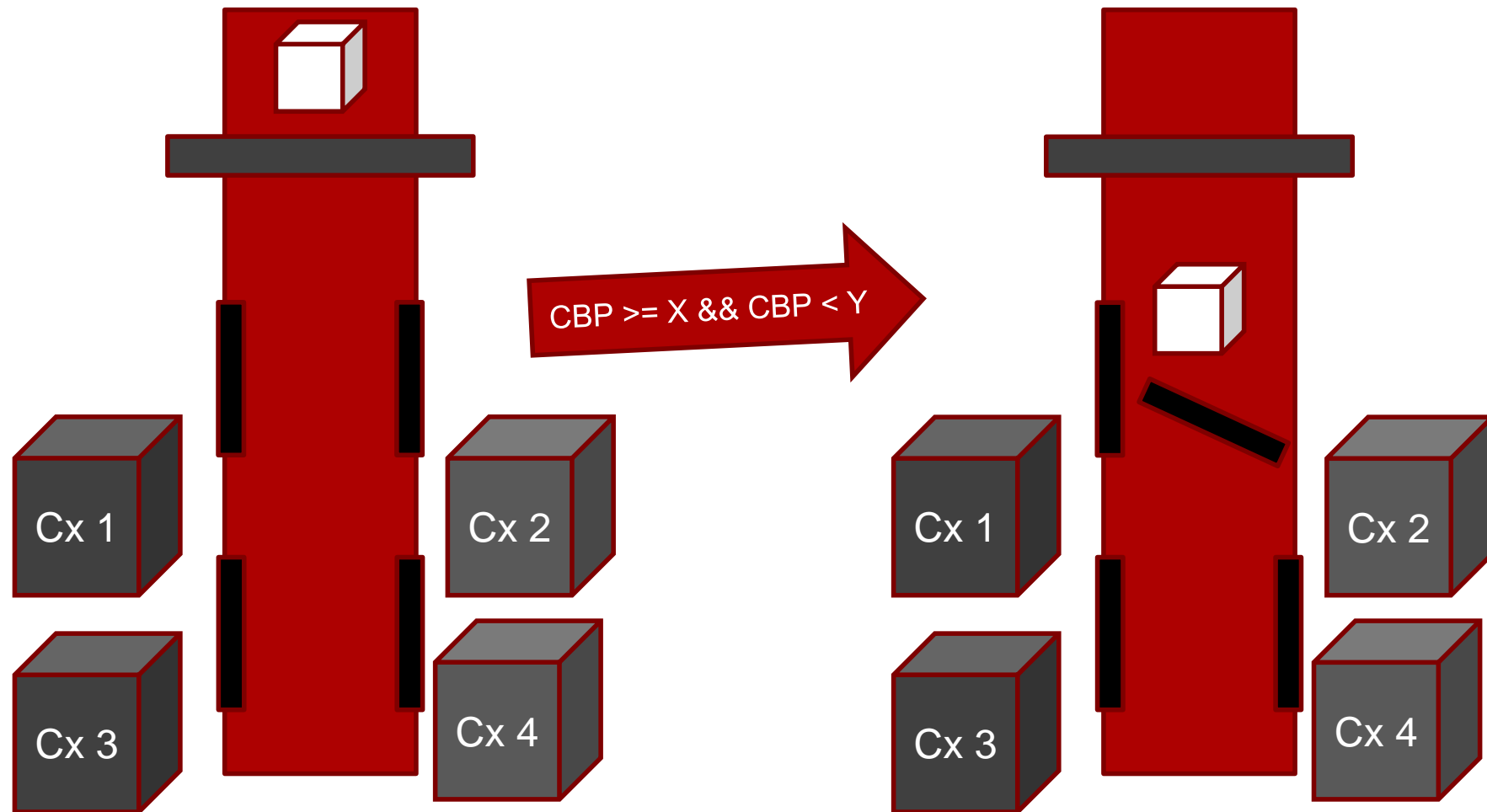
```
caso '/':  se y <> 0 então
           início
               resultado ← x / y;
               escreva("Divisão = ", resultado);
           fim
           senão
           início
               escreva("Não existe divisão por ZERO");
           fim se;
caso '%':  se y <> 0 então
           início
               resultado ← x mod y;
               escreva("Resto da Divisão = ", resultado);
           fim
           senão
           início
               escreva("Não existe divisão por ZERO");
           fim se;
caso contrário: escreva("Opção Inválida");
fim escolha;
Fim.
```

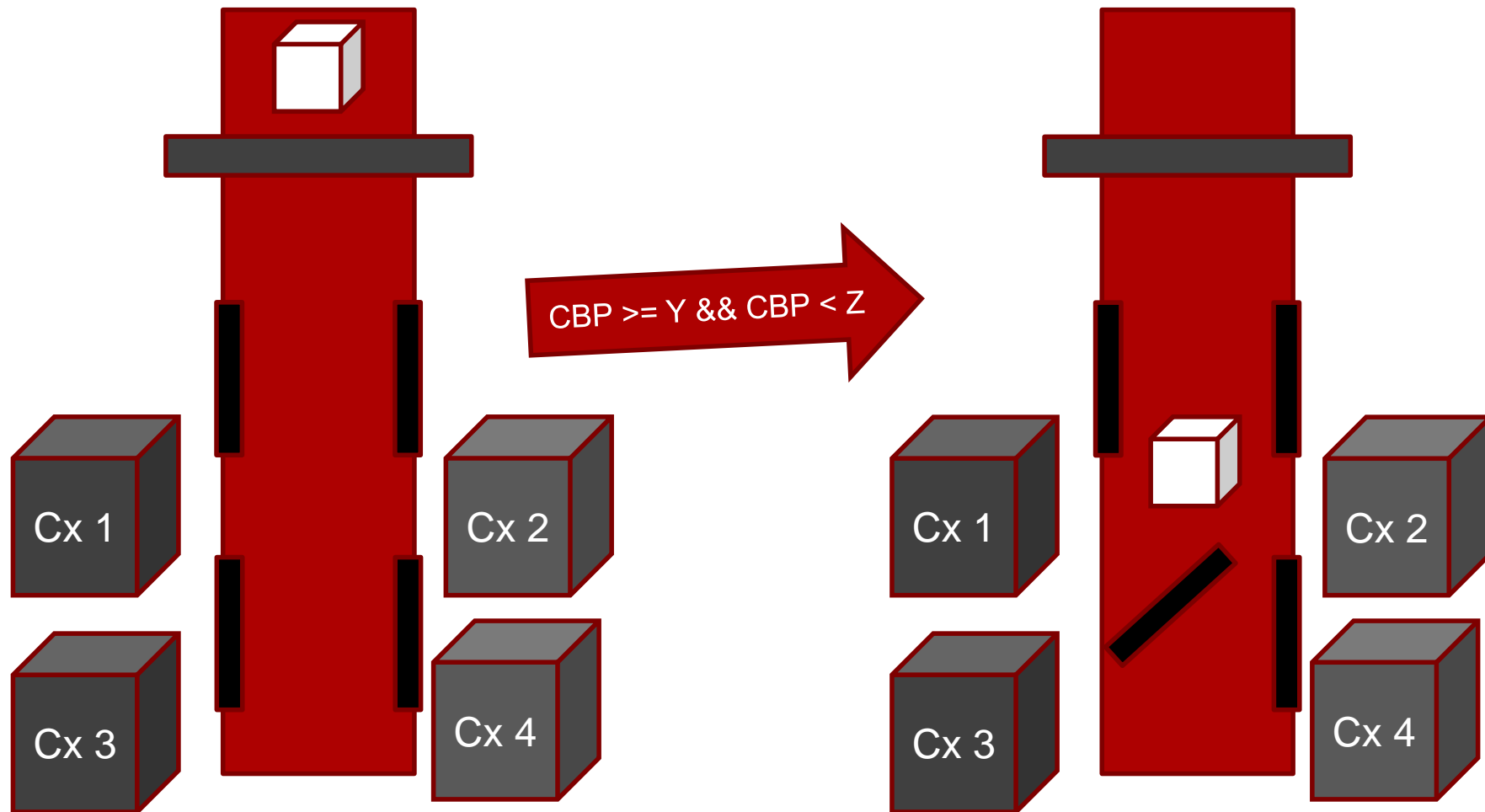


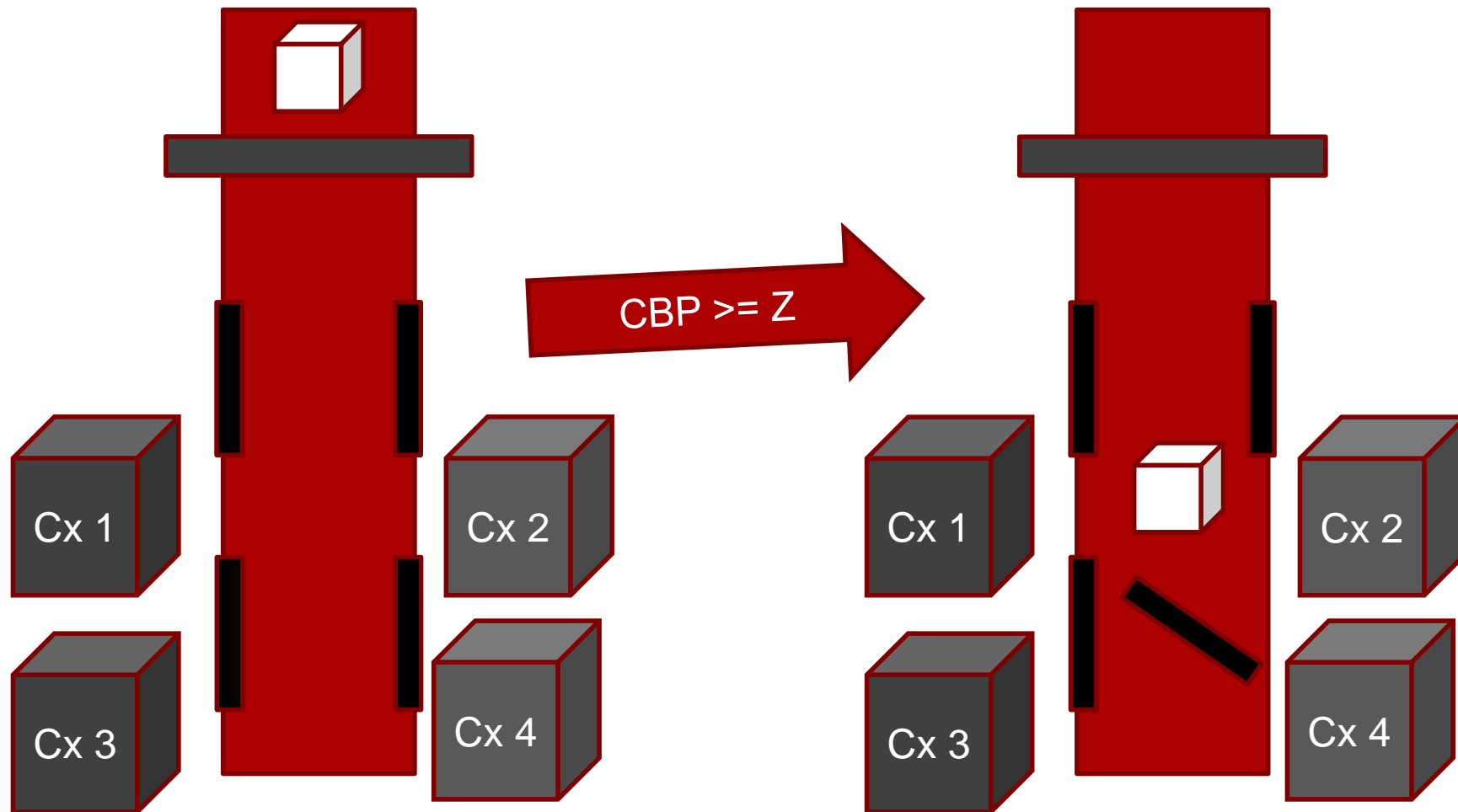
AULA DE ALGORITMO ESTRUTURA DE DECISÃO ENCAIXADA

Profa. M. Sc. Valéria Maria Volpe









Estruturas de Decisão

- **Estrutura de decisão encaixada**

- Esta estrutura deve ser usada quando, devido a necessidade de processamento, agrupamos várias estrutura de decisão encaixada (aninhada).
- Ocorre quando uma determinada ação (bloco de comandos) deve ser executada se um conjunto de possibilidades ou combinações de situações forem satisfeitas (verdadeiro) ou não (falso).

Estruturas de Decisão

- Faça um algoritmo/programa para ler a idade de um nadador, classifique-o de acordo com as categorias.

| Idade | Categoria |
|------------------|------------|
| De 5 a 7 anos | Infantil A |
| De 8 a 10 anos | Infantil B |
| De 11 a 13 anos | Juvenil A |
| De 14 a 17 anos | Juvenil B |
| Maior de 17 anos | Adulto |

- Ações:
 - Ler idade
 - Classificar de acordo com a tabela
- Variáveis
 - Idade - inteiro

Estruturas de Decisão

Algoritmo Nadadores;

Inicio

var

inteiro: idade;

leia (idade);

se idade \geq 5 E idade \leq 7 então

inicio

escreva (“Categoria – Infantil A”);

fim

senão

inicio

Estruturas de Decisão

se idade \geq 8 E idade \leq 10 então

início

escreva (“Categoria – Infantil B”);

fim

senão

início

se idade \geq 11 E idade \leq 13 então

início

escreva (“Categoria – Juvenil A”);

fim

senão

início

Estruturas de Decisão

se idade \geq 14 e idade \leq 17 então

início

 escreva (“Categoria – Juvenil B”);

fim

senão

início

se idade \geq 18 então

início

 escreva (“Categoria – Adulto”);

fim

senão

início

Estruturas de Decisão

Escreva (“Você não pode competir”)

fim se;

fim se;

fim se;

fim se;

fim se ;

Fim.

Estruturas de Decisão

```
#include <stdio.h>
#include <stdlib.h>
main()
{
    int idade;
    printf ("\n Digite sua idade: ");
    scanf("%d", & idade);
    if(idade >= 5 && idade <= 7)
    {
        printf ("\n Categoria – Infantil A \n");
    }
    else
    {
```


Estruturas de Decisão

```
if (idade >= 8 && idade <= 10)
{
    printf ("\n Categoria – Infantil B \n");
}
else
{
    if (idade >= 11 && idade <= 13)
    {
        printf ("\n Categoria – Juvenil A \n");
    }
    else
    {
```

Estruturas de Decisão

```
if (idade >= 14 && idade <= 17)
{
    printf ("\n Categoria – Juvenil B \n");
}
else
{
    if (idade >= 18)
    {
        printf ("\n Categoria – Adulto \n");
    }
    else
    {
```

Estruturas de Decisão

```
printf ("\n Você não pode competir \n")
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

Estruturas de Decisão

- Exercícios:
 1. Faça um algoritmo para ler três números inteiros e mostrar na tela o maior número digitado.
 2. Faça um algoritmo para ler 3 números reais e verificar se é possível formar um triângulo ou não. Se formar um triângulo verifique que tipo de triângulo é formado: equilátero (três lados iguais), isósceles (dois lados iguais e um diferente) ou escaleno (três lados diferentes). Sabendo que para formar um triângulo é necessário que:

$Lado_A < Lado_B + Lado_C$ E $Lado_B < Lado_A + Lado_C$ E $Lado_C < Lado_A + Lado_B$

3. Faça um algoritmo para ler três números inteiros e colocá-los em ordem crescente.

Estruturas de Decisão

Algoritmo Maior Número;

Início

var

inteiro: a, b, c;

leia(a);

leia(b);

leia(c);

se a \geq b E a \geq c então

início

escreva("Maior Número: ", a);

fim

senão

início

se b \geq a E b \geq c então

início

escreva("Maior Número: ", b);

fim

senão

início

escreva("Maior Número: ", c);

fim se;

fim se;

Fim.

Estruturas de Decisão

Algoritmo Triângulo;

Início

var

real: ladoA, ladoB, ladoC;

leia(ladoA);

leia(ladoB);

leia(ladoC);

se ladoA < ladoB + ladoC E ladoB < ladoA + ladoC E ladoC < ladoA + ladoB então

início

escreva("Forma um triângulo");

se ladoA = ladoB E ladoB = ladoC então

início

escreva("Equilátero");

fim

senão

início

se ladoA = ladoB OU ladoA = ladoC OU ladoB = ladoC então

início

escreva("Isósceles");

fim

senão

início

escreva("Escaleno");

fim se;

fim se;

fim

senão

início

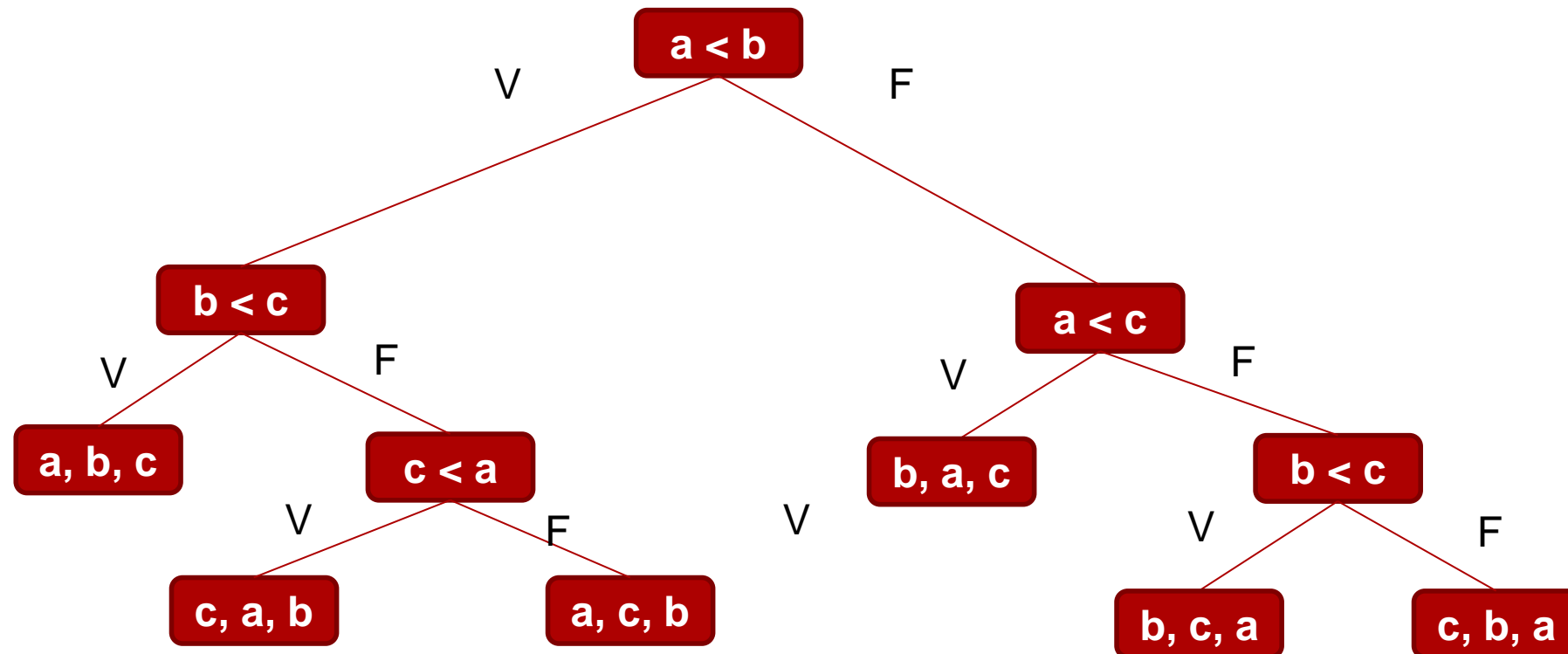
escreva("Não forma um Triângulo ");

fim se;

Fim.

Estrutura de Decisão

- Faça um algoritmo e programa em **LINGUAGEM C** para ler três números inteiros e colocá-los em ordem crescente.



Solução Exercício 3

Algoritmo Ordem Crescente;

Início

var

inteiro: a, b, c;

leia(a);

leia(b);

leia(c);

se a < b então

início

se b < c então

início

escreva("Ordem Crescente: ", a, b, c);

fim

senão

início

se c < a então

início

escreva("Ordem Crescente: ", c, a, b);

fim

Solução Exercício 3

```

    senão
    início
        escreva("Ordem Crescente: ", a, c, b);
    fim se;
fim se;
fim
senão
início
    se a < c então
    início
        escreva("Ordem Crescente: ", b, a, c);
    fim
    senão
    início
        se b < c então
        início
            escreva("Ordem Crescente: ", b, c, a);
        fim
    fim
fim
```

Solução Exercício 3

```
        senão  
        início  
            escreva("Ordem Crescente: ", c, b, a);  
        fim se;  
    fim se;  
fim se;  
Fim.
```

Estruturas de Decisão

- Exercícios:
 4. Faça um algoritmo que leia a data de nascimento de uma pessoa, ou seja dia, mês e ano, leia o dia, o mês e o ano atual e calcule a idade correta da pessoa. Verifique se ela está fazendo aniversário e mostre na tela a mensagem “Parabéns a você nesta data querida!!!”.
 5. O custo ao consumidor de um carro novo é a soma do custo de fábrica mais a porcentagem do distribuidor e mais impostos, ambos aplicados ao custo de fábrica. Sabe-se que as porcentagens e impostos são os que estão na tabela abaixo. Faça um algoritmo que leia o custo de fábrica de um carro e mostre o custo ao consumidor.

| Preço de Custo | Porcentagem do Distribuidor | imposto |
|--------------------------------|-----------------------------|---------|
| Até R\$ 12000,00 | 5% | Isento |
| De R\$ 12000,01 a R\$ 25000,00 | 10% | 7,5% |
| Acima de R\$ 25000,00 | 25% | 10,5% |

Solução Exercício 4

Algoritmo Parabéns;

Início

var

inteiro: diaNasc, mesNasc, anoNasc, diaAtual, mesAtual, anoAtual, idade;

leia(diaNasc);

leia(mesNasc);

leia(anoNasc);

leia(diaAtual);

leia(mesAtual);

leia(anoAtual);

idade \leftarrow anoAtual – anoNasc;

// Continua no próximo slide

Solução Exercício 4

```
se mesNasc = mesAtual então
início
    se diaNasc = diaAtual então
    início
        escreva("Parabéns a você nesta data querida... ");
    fim
    senão
    início
        se diaNasc > diaAtual então
        início
            idade ← idade - 1;
        fim
    fim se;
fim
```

// Continua no próximo slide

Solução Exercício 4

senão

início

se mesNasc > mesAtual então

início

idade \leftarrow idade - 1;

fim se;

fim se;

escreva("Idade = ", idade);

Fim.

Exercícios

- 1) Faça um algoritmo que leia o valor dos catetos de um triângulo, calcule e mostre a hipotenusa.
- 2) Faça um algoritmo que leia a temperatura em graus Celsius, calcule e mostre a temperatura em graus Fahrenheit. Sabendo que: **$Fahrenheit = 1.8 * Celsius + 32$** .
- 3) O custo ao consumidor de um carro novo é a soma do custo de fábrica com a porcentagem do distribuidor e com os impostos, ambos aplicados ao custo de fábrica. Sabe-se que as porcentagens são as que estão na tabela abaixo. Faça um algoritmo que leia o custo de fábrica de um carro e mostre o custo ao consumidor.

| Custo de Fábrica | Porcentagem do consumidor | Impostos |
|------------------------|---------------------------|----------|
| Até R\$ 12.000,00 | 5% | Isento |
| Acima de R\$ 12.000,00 | 10% | 15% |

Exercícios

- 4) As maçãs custam R\$ 1,30 cada se forem compradas menos do que uma dúzia, e R\$ 1,25 se forem compradas pelo menos doze. Escreva um algoritmo que leia o número de maçãs compradas, calcule e escreva o valor total da compra

Solução – Exercício 1

• Algoritmo Hipotenusa;

Início

var

real: cateto1, cateto2, hipotenusa;

leia(cateto1);

leia(cateto2);

$hipotenusa \leftarrow \sqrt{cateto1^2 + cateto2^2};$

escreva("Hipotenusa = ", hipotenusa);

Fim.

Solução – Exercício 2

Algoritmo Celsius Fahrenheit;

Início

var

real: celsius, fahrenheit;

leia(celsius);

fahrenheit \leftarrow 1.8 * celsius + 32;

escreva("Celsius = ", celsius, "Fahrenheit = ", fahrenheit);

Fim.

Solução – Exercício 3

Algoritmo Valor Carro;

Início

var

real: custoCarro, percentConsumidor, impostos, valorCarro;

leia(custoCarro);

se custoCarro \leq 12000.00 então

início

percentConsumidor \leftarrow custoCarro * 0.05;

impostos \leftarrow 0.0;

fim

senão

início

percentConsumidor \leftarrow custoCarro * 0.10;

impostos \leftarrow custoCarro * 0.15;

fim se;

valorCarro \leftarrow custoCarro + percentConsumidor + impostos;

escreva("Valor do Carro R\$ ", valorCarro);

Fim.

Exercício para entregar

- Uma confeitaria está vendendo doces com a seguinte tabela de preços:
Escreva um **ALGORITMO** para ler o tipo e a quantidade (em Kg) de doce. Calcule e mostre o valor a ser pago pelo cliente.

| Tipo da doce | Doce | Até 5 Kg | Acima de 5 Kg |
|--------------|-------|------------------|------------------|
| 1 | Trufa | R\$ 25,00 por Kg | R\$ 22,00 por Kg |
| 2 | Torta | R\$ 24,45 por Kg | R\$ 22,25 por Kg |
| 3 | Bolo | R\$ 35,00 por Kg | R\$ 30,00 por Kg |

Bibliografia

- **Básica**

ASCENCIO, A. F. G, CAMPOS, E. A. V. **Fundamentos da Programação de Computadores**: algoritmos, Pascal e C/C++ e Java. Longman, 2007.

FORBELLONE, L. V., EBERSPACHER, H. F. **Lógica de Programação**: a construção de algoritmos e estruturas de dados. Prentice Hall, 2005.

ZIVIANI, Nivio. **Projeto de Algoritmos com Implementações em Pascal e C**. 2.ed. Thomson Pioneira, 2004.

- **Complementar**

FARRER, H et al. **Algoritmos estruturados**. 3 ed. Rio de Janeiro: LTC, 1999. 284 p.

MANZANO, J. A. N. G.; **Estudo dirigido de algoritmos**. 9. ed. São Paulo: Érica, 2004.

LOUNDON, L. **Algoritmos em C**. São Paulo: Ciência Moderna, 2000.

ASCENCIO, A. F. G.; CAMPOS, E.A.V. **Fundamentos da programação de computadores**: algoritmo, pascal e C++. São Paulo: Pearson Prentice Hall, 2002. 355 p