

AULA DE ALGORITMO - 02

Profa. M. Sc. Valéria Maria Volpe

Estruturas de Controle

2

- 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, necessitamos 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 Controla

3

□ **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

4

□ 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

5

□ 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

6

□ Sintaxe – Algoritmo

se **Condição** então

início **V**

→ Bloco de comandos que será executado somente

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

F

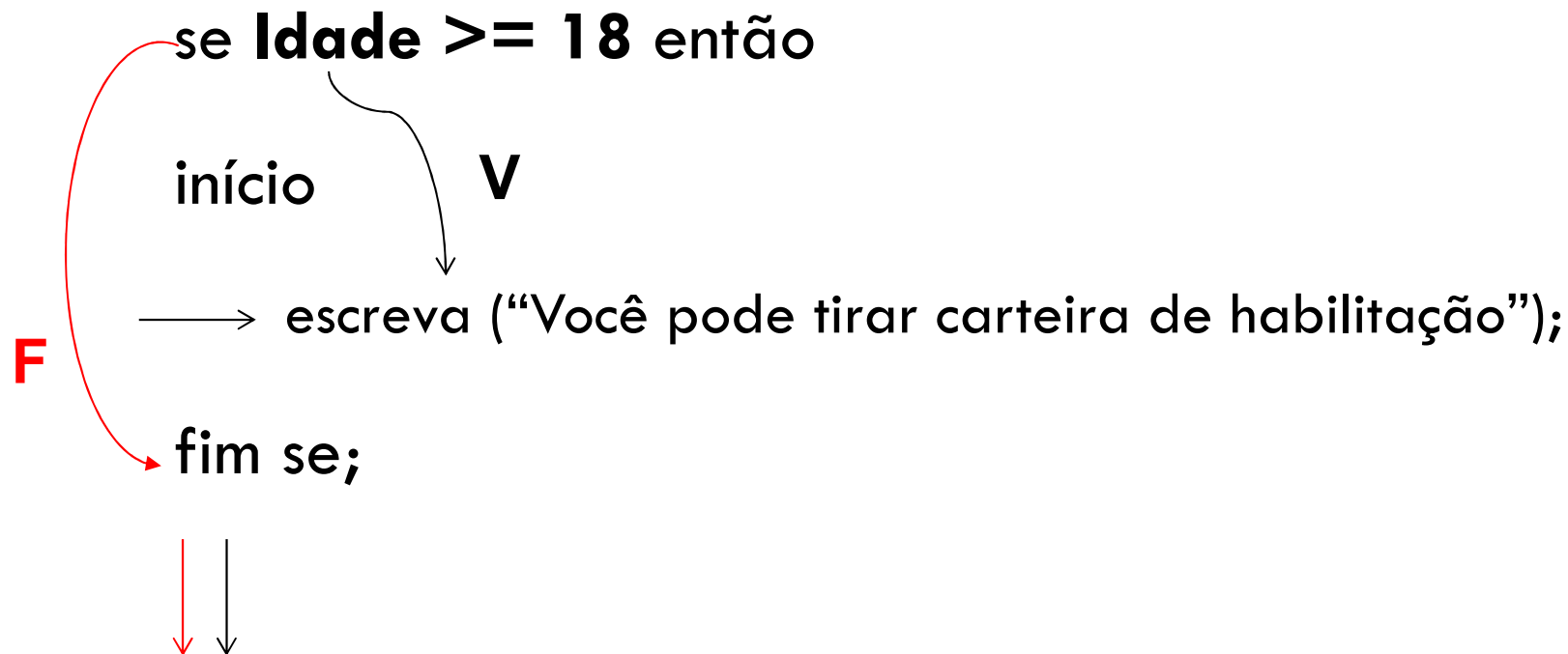
fim se;



Estruturas de Decisão

7

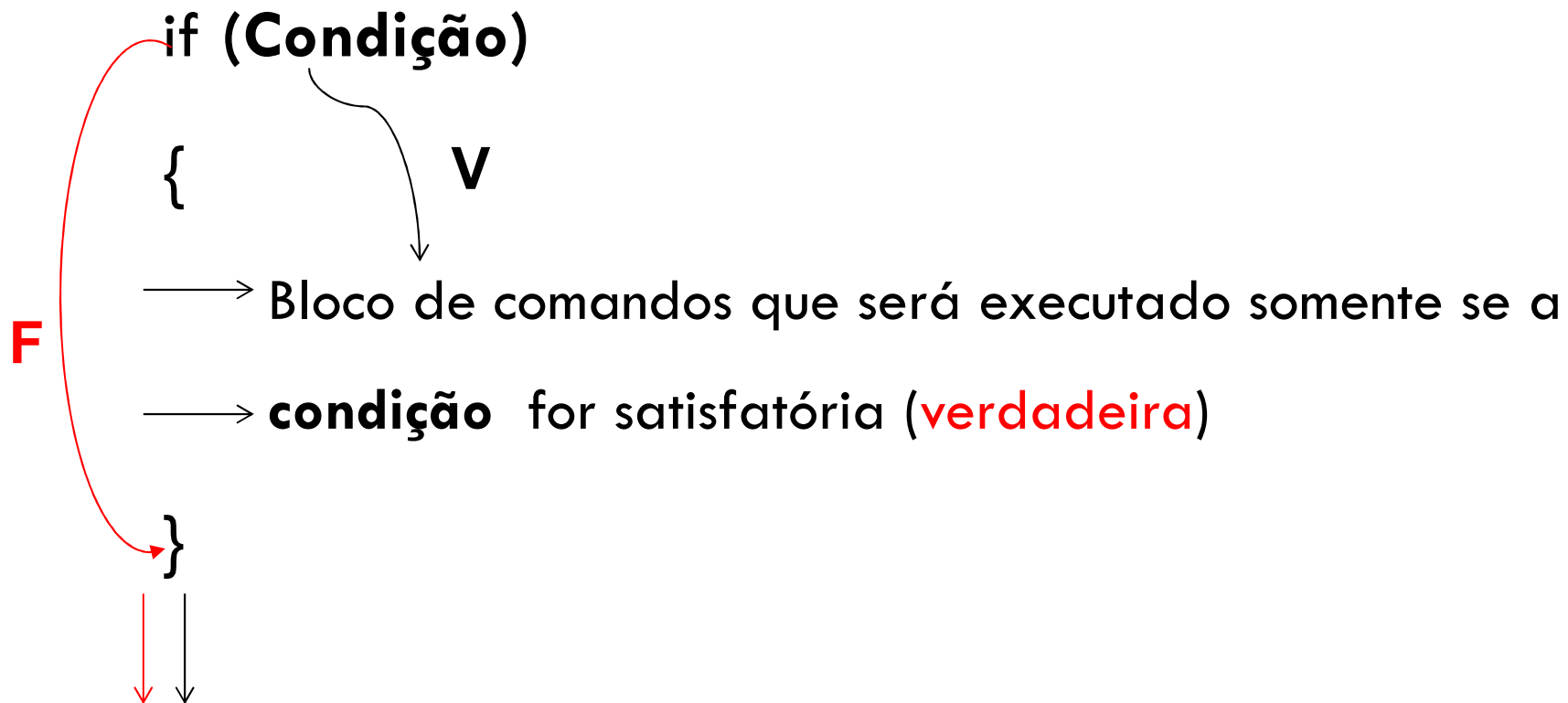
□ Exemplo:



Estruturas de Decisão

8

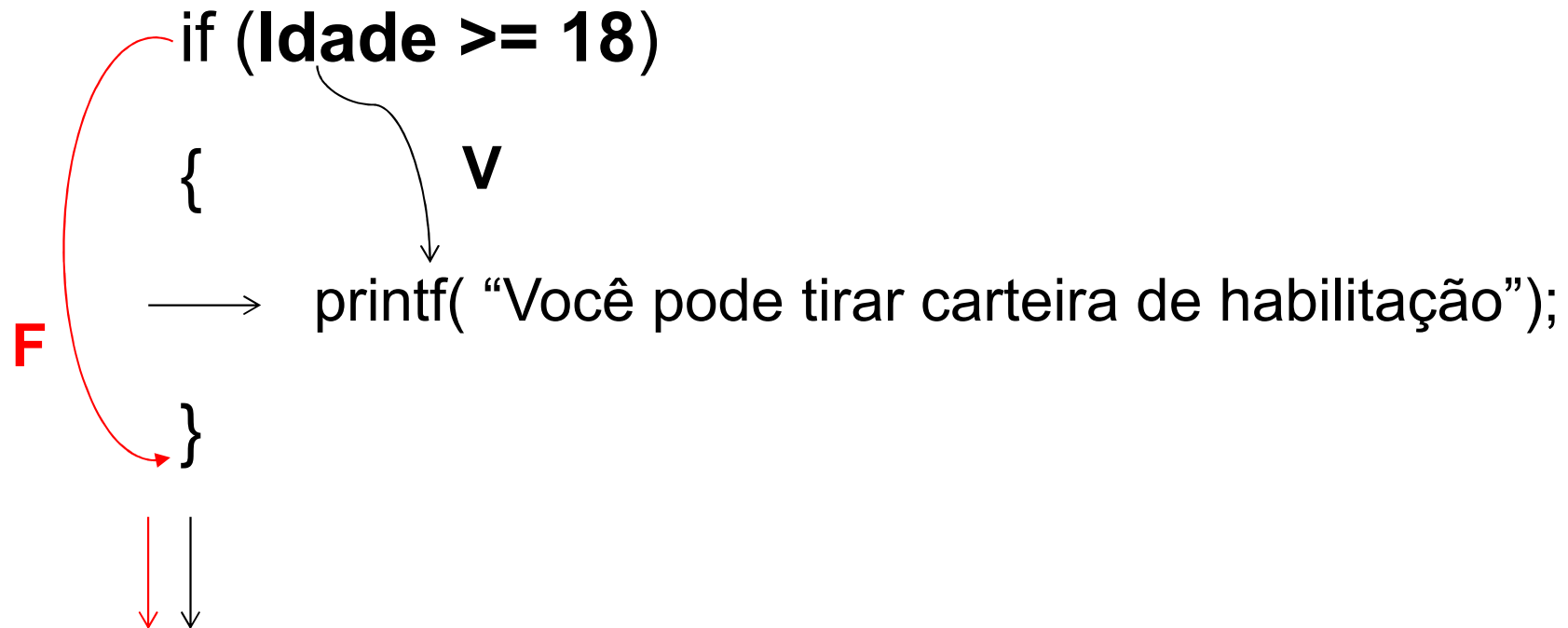
□ Sintaxe – Linguagem C



Estruturas de Decisão

9

□ Exemplo:



Estruturas de Decisão

10

□ 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

11

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

12

```
#include<stdio.h>
#include<stdlib.h>
main ()
{
    int Num, Resto;
    printf("\n Digite um n° 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

13

□ 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.

Solução Exercício 2

14

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");

Fim.

Solução Exercício 3

15

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, " pontos");

Fim.

Solução 1 Exercício 4

16

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

17

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.

Estruturas de Decisão

18

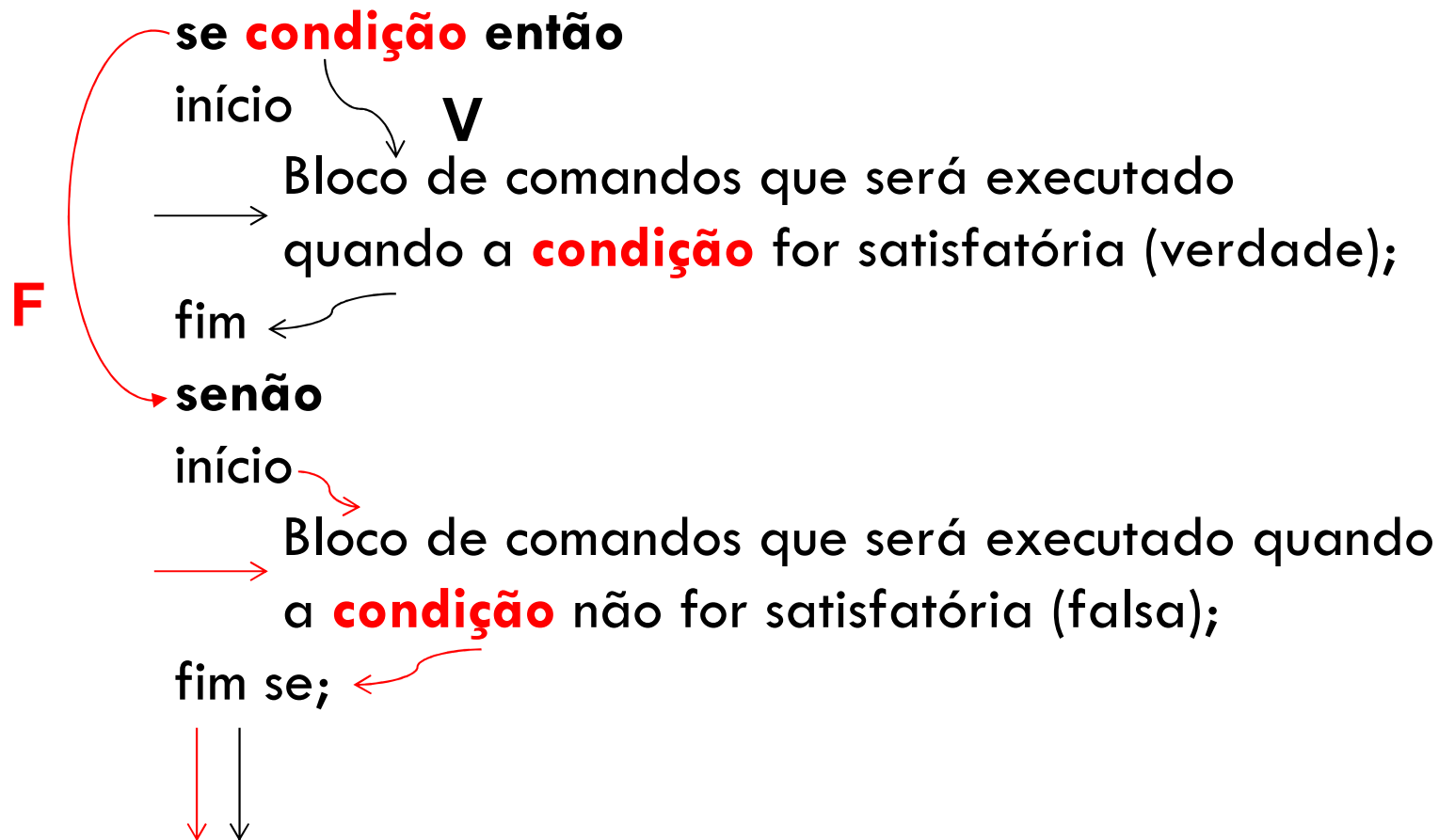
□ Estrutura de Decisão Composta

- 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

19

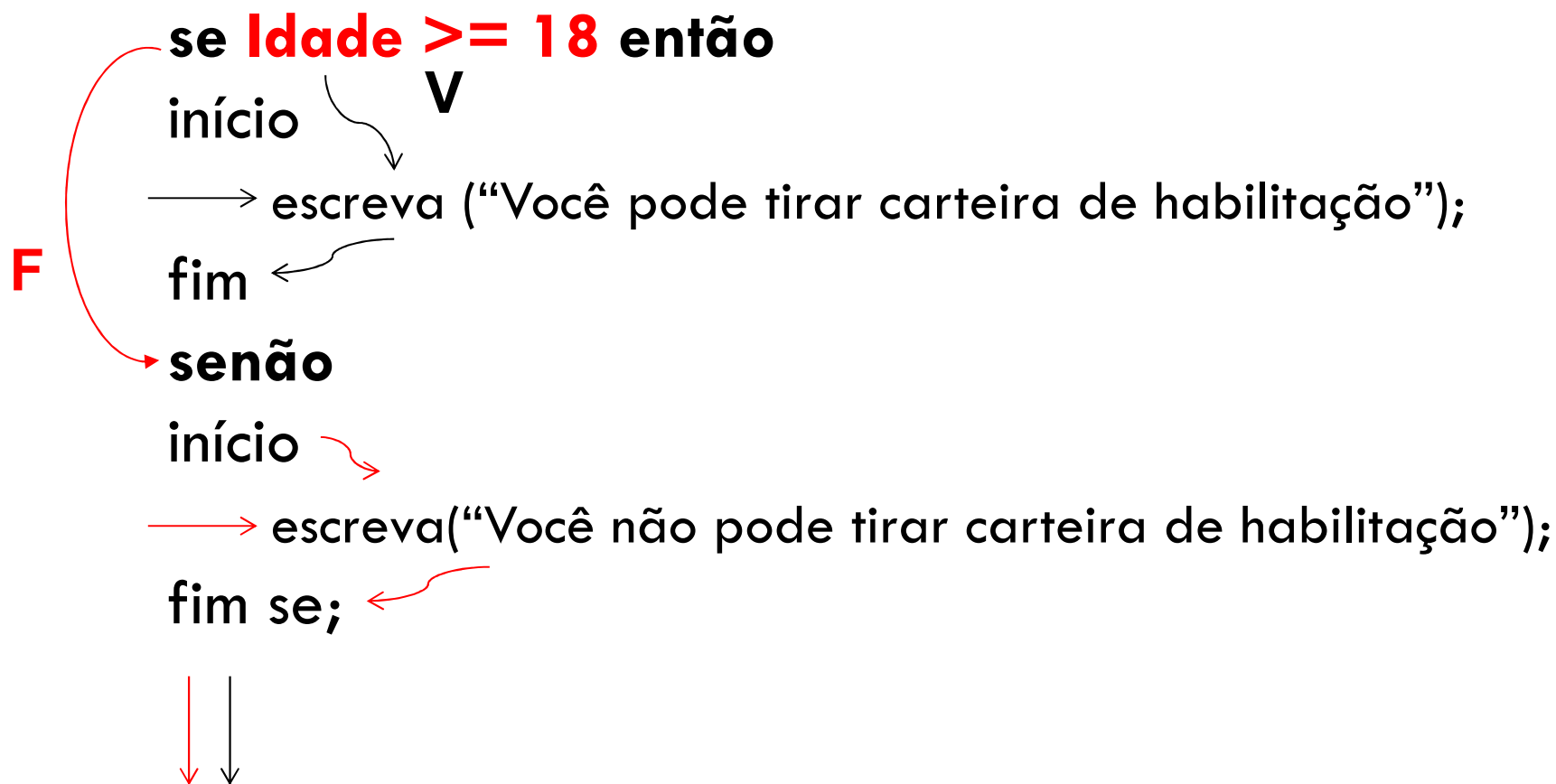
□ Sintaxe – Algoritmo



Estruturas de Decisão

20

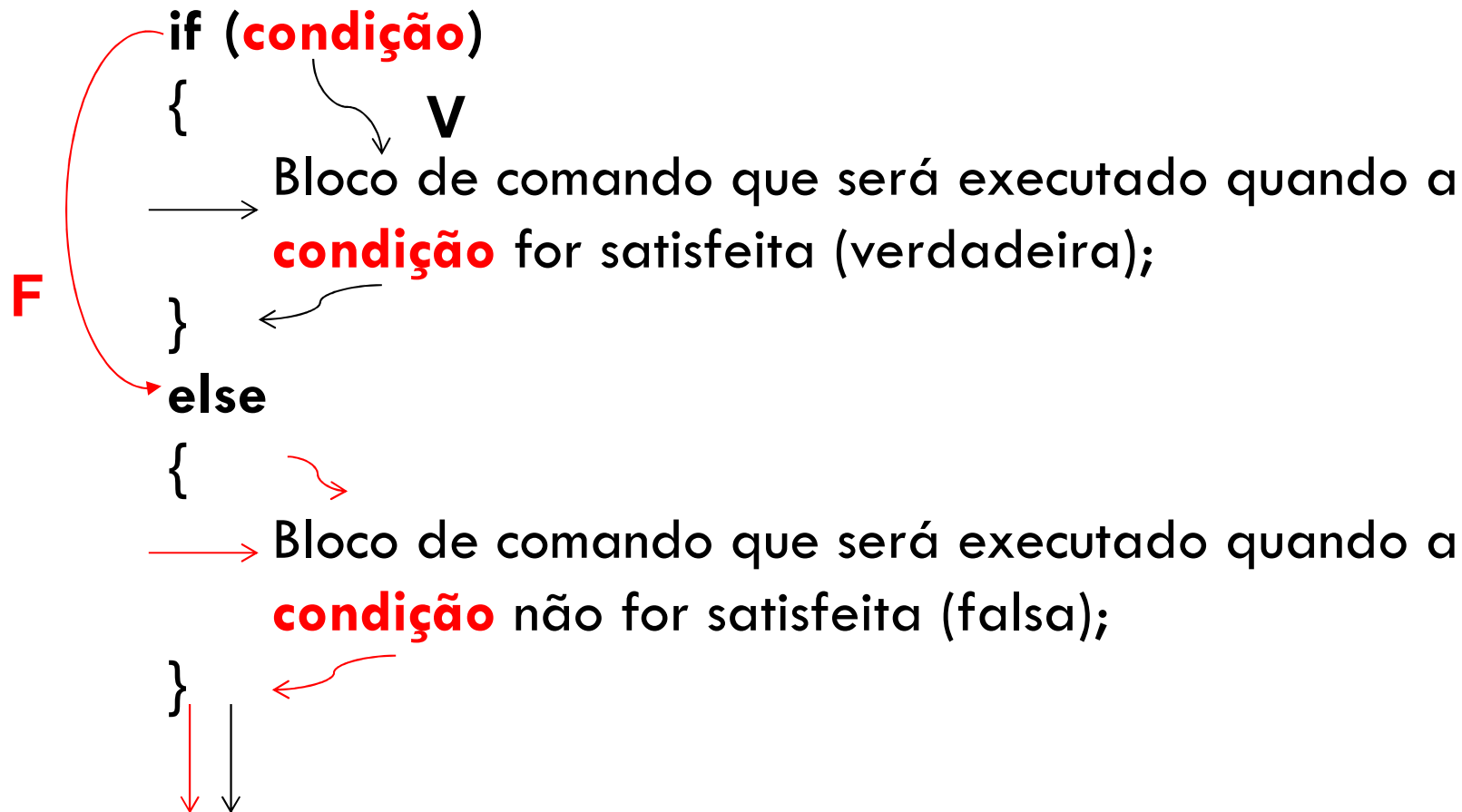
□ Exemplo:



Estruturas de Decisão

21

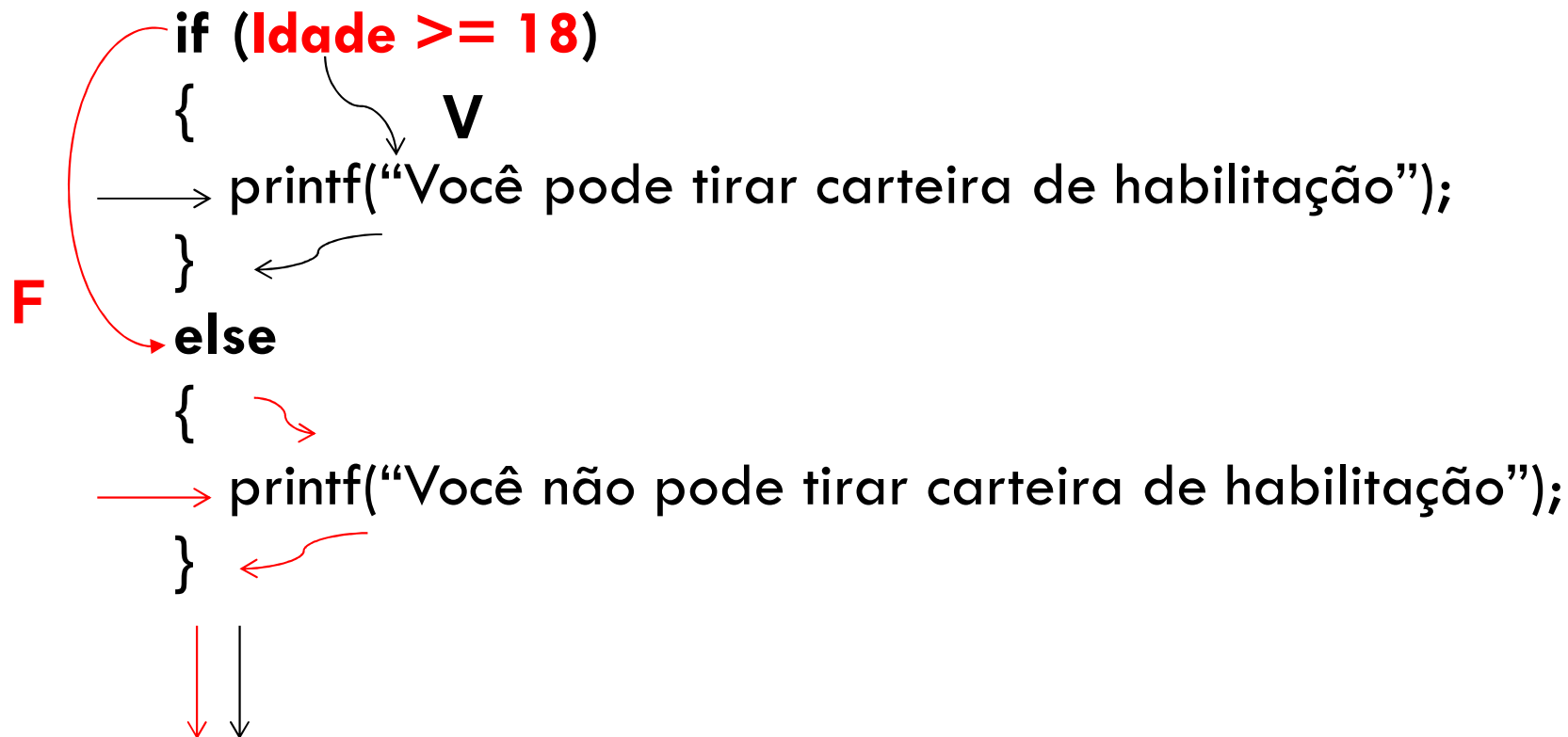
□ Sintaxe – Linguagem C



Estruturas de Decisão

22

□ Exemplo:



Estruturas de Decisão

23

□ **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

24

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

25

```
#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;
    if (Idade >= 16 )
    {
        printf("\n Você pode votar!! \n");
    }
    else
    {
        printf("\n Você não pode votar \n");
    }
    system ("pause");
}
```

Estruturas de Decisão

26

- 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:
 - Homem: $(72.7 * altura) - 58.0$
 - Mulher: $(62.1 * altura) - 44.7$
- Ações:
 - Ler altura e sexo
 - Verificar o sexo M (Masculino) ou F (Feminino)
 - Mostrar o resultado
- Variáveis:
 - Altura
 - Sexo
 - Pesoldeal

Estruturas de Decisão

27

Algoritmo Peso Ideal;

Início

var

real: Altura, Pesoldeal;

caracter: Sexo;

leia (Altura);

leia (Sexo);

se Sexo = 'M' ou Sexo = 'm' então

início

$\text{Pesoldeal} \leftarrow (72.7 * \text{Altura}) - 58.0;$

fim

senão

início

$\text{Pesoldeal} \leftarrow (62.1 * \text{Altura}) - 44.7;$

fim se;

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

Fim.

Estruturas de Decisão

28

```
#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 == 'M' || Sexo == 'm' )
    {
        Pesoldeal = (72.7 * Altura) - 58.0;
    }
    else
    {
        Pesoldeal = (62.1 * Altura) - 44.7;
    }
    printf("\n Seu peso Ideal é: %.2f Kg", Pesoldeal);
    system ("pause");
}
```

Estruturas de Decisão

29

□ 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.

Solução Exercício 1

30

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 Exercício 2

31

Algoritmo Aumento Preço;

Início

var

inteiro: codigodepto;

real: valorproduto, preconovo;

leia (valorproduto);

leia (codigodepto);

se codigodepto = 1 então

início

preconovo \leftarrow valorproduto + valorproduto * 0.05;

fim

senão

início

preconovo \leftarrow valorproduto + valorproduto * 0.074;

fim se;

escreva("Preço Novo: ", preconovo);

Fim.

Estruturas de Decisão

32

- **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

33

□ 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

34

- **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

35

□ Exemplo

escolha (X)

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

36

□ Sintaxe – Linguagem C

switch (**NomeVar**)

{

case v1: Bloco de comandos que será executado quando **NomeVar = v1**;
 break;

case v2: Bloco de comandos que será executado quando **NomeVar = v2**;
 break;

.
.
.

case vn: Bloco de comandos que será executado quando **NomeVar = vn**;
 break;

default: Bloco de comandos que será executado quando **NomeVar ≠ v1, v2, ..., vn**;

}

Estruturas de Decisão

37

□ Exemplo

switch (X)

{

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

38

□ 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

39

□ 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

40

□ 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

41

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;

ALGORITMO - VMV - 2016

Fim.

Estruturas de Decisão

42

```
#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

43

□ 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

44

Algoritmo 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

45

```
#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

46

Algoritmo Operações;

Início

var

real: x, y, resultado;

inteiro: codigo;

leia(x);

leia(y);

leia(codigo);

escolha (codigo)

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);

Estruturas de Decisão

47

```
caso 3: resultado  $\leftarrow$  x * y;  
        escreva("Multiplicação = ", resultado);  
caso 4: se y  $\neq$  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

48

□ 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

49

□ Exemplo

se condição 1 então

início

Comando1,

se condição2 então

início

Comando2;

fim

senão

início

se condição3 então

início

Comando3

fim se;

Comando4,

fim se;

fim

Estruturas de Decisão

50

senão

inicio

se condição4 então

inicio

Comando5

fim

senão

inicio

escolha (variável)

caso v1: comando6

caso v2: comando7

caso v3: comando8

caso v4: comando9

caso contrario: comando10

fim escolha;

fim se;

fim se;

Comando11

Estruturas de Decisão

51

ANÁLISE DA ESTRUTURA DE DECISÃO
ENCAIXADA

CONDIÇÕES				VALORES – ESCOLHA					COMANDOS QUE SERÃO
1	2	3	4	V1	V2	V3	V4	CC	EXECUTADOS
V	V	-	-	-	-	-	-	-	1, 2, 11
V	F	V	-	-	-	-	-	-	1, 3, 4, 11
V	F	F	-	-	-	-	-	-	1, 4, 11
F	-	-	V	-	-	-	-	-	5, 11
F	-	-	F	V	-	-	-	-	6, 11
F	-	-	F	-	V	-	-	-	7, 11
F	-	-	F	-	-	V	-	-	8, 11
F	-	-	F	-	-	-	V	-	9, 11
F	-	-	F	-	-	-	-	V	10, 11

Estruturas de Decisão

52

- 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

53

Algoritmo Nadadores;

Início

var

inteiro: Idade;

leia (Idade);

se Idade \geq 5 E Idade \leq 7 então

início

escreva (“Categoria – Infantil A”);

fim

senão

início

Estruturas de Decisão

54

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

55

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

56

```
        Escreva ("Você não pode competir")
    fim se;
    fim se;
    fim se;
    fim se;
    fim se ;
Fim.
```


Estruturas de Decisão

57

```
#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

58

```
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

59

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

Estruturas de Decisão

60

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

```
}  
}  
}  
}  
}  
}
```

Estruturas de Decisão

61

□ 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**

Estruturas de Decisão

Algoritmo Maior Número;

62

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

63

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.

ALGORITMO - VMV - 2016

Estruturas de Decisão

64

□ Exercícios:

3. 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!!!”.
4. 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 3

65

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;

se mesnasc = mesatual então

início

se dianasc = diaatual então

início

escreva("Parabéns a você nesta data querida... ");

fim

senão

Solução Exercício 3

66

```
início
    se dianasc > diaatual então
        início
            idade ← idade - 1;
        fim
    fim se;
fim
senão
início
    se mesnasc > mesatual então
        início
            idade ← idade - 1;
        fim se;
    fim se;
escreva("Idade = ", idade);
Fim.
```

Bibliografia

67

□ 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