

CONSTRUÇÃO DE ALGORITMOS

Bacharelado em Sistemas da Informação

Prof. Marco André Abud Kappel

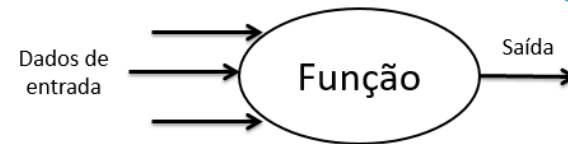
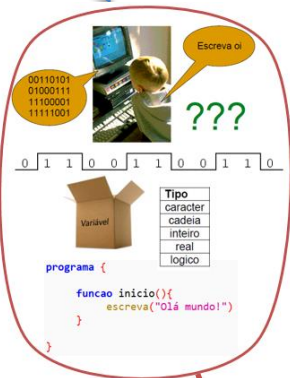
Aula 6 – Estruturas de decisão

Estruturas de decisão

- Anteriormente:



Hoje:
Estruturas de
decisão



```
funcao tipo_de_retorno identificador(tipo1 par1, tipo2 par2, ...){  
    bloco de comandos  
    retorne algum_resultado  
}
```

Parâmetros por valor

Parâmetros por referência

falso
verdadeiro



nao e ou

Operador	Operação
==	igual
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual

x	y	x e y
falso	falso	falso
falso	verdadeiro	falso
verdadeiro	falso	falso
verdadeiro	verdadeiro	verdadeiro

x	y	x ou y
falso	falso	falso
falso	verdadeiro	verdadeiro
verdadeiro	falso	verdadeiro
verdadeiro	verdadeiro	verdadeiro

Estruturas de decisão

- **Introdução**

- Todos os nossos programas, até o momento, seguem uma **linha sequencial** de execução.
- Ou seja, todas as linhas do algoritmo estão sendo executadas **uma após a outra**.



- Porém, a maior parte dos algoritmos exige uma **complicação** a mais.

Estruturas de decisão

- **Introdução**

- Vamos relembrar um dos primeiros algoritmos que vimos: trocar uma lâmpada.

```
1 - Acionar o interruptor
2 - Se a lâmpada acender, encerre.
3 - Pegar uma escada
4 - Posicionar a escada embaixo da lâmpada
5 - Buscar uma lâmpada nova
6 - Subir na escada
7 - Retirar a lâmpada velha
8 - Colocar a lâmpada nova
9 - Descer da escada
10 - Se a lâmpada nova acender, encerre.
11 - Repita os passos 5 a 10.
```



Estruturas de decisão

- **Introdução**

- Vamos relembrar um dos primeiros
uma lâmpada.

```
1 - Acionar o interruptor
2 - Se a lâmpada acender, encerre.
3 - Pegar uma escada
4 - Posicionar a escada embaixo da lâmpada
5 - Buscar uma lâmpada nova
6 - Subir na escada
7 - Retirar a lâmpada velha
8 - Colocar a lâmpada nova
9 - Descer da escada
10 - Se a lâmpada nova acender, encerre.
11 - Repita os passos 5 a 10.
```



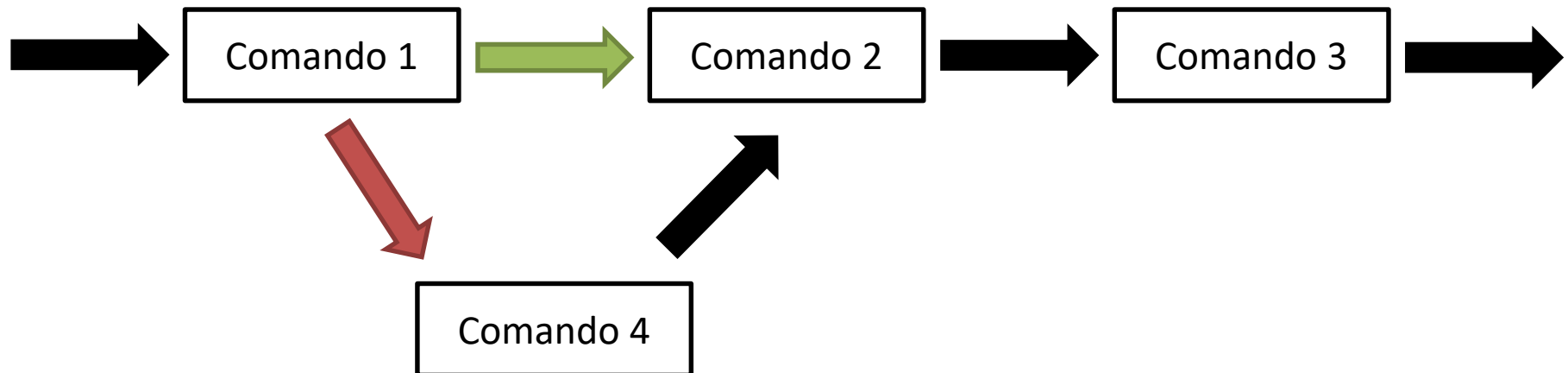
Perceba que, mesmo sendo muito simples, este algoritmo possui alguns desvios condicionais.



Estruturas de decisão

- **Desvio condicional**

- A **mesma lógica** será aplicada em nossos programas, daqui em diante.
- Isso significa que alguns comandos **podem ou não** ser executados, dependendo de uma **condição**.



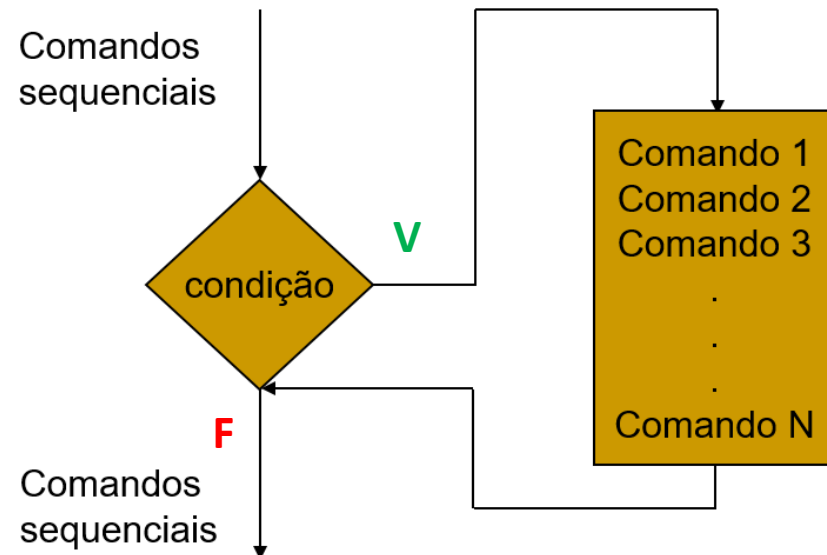
Estruturas de decisão

- **Desvio condicional**

- O programa funciona normalmente, até atingir uma **condição**.
- A condição é uma **expressão lógica**, ou seja, retorna um **valor lógico**.

- Por exemplo:

- A pessoa é maior de idade?
- O produto está na validade?
- A senha inserida é a correta?
- O aluno está aprovado?



Estruturas de decisão

- **Comando se**

- Em Portugol, se desejarmos que um determinado conjunto de comandos seja executado **apenas** se uma condição for satisfeita, devemos usar a seguinte sintaxe:


```
se (condição){  
    comando1  
    comando2  
    comando3  
    ...  
}
```


Estruturas de decisão

- **Comando se**

- Em Portugal, se desejarmos que um determinado conjunto de comandos seja executado **apenas** se uma condição for satisfeita, devemos usar a seguinte sintaxe:

Palavra reservada que indica um desvio condicional.



```
se (condição){  
    comando1  
    comando2  
    comando3  
    ...  
}
```

Estruturas de decisão

- **Comando se**

- Em Portugal, se desejarmos que um determinado conjunto de comandos seja executado **apenas** se uma condição for satisfeita, devemos usar a seguinte sintaxe:

Palavra reservada que indica um desvio condicional.

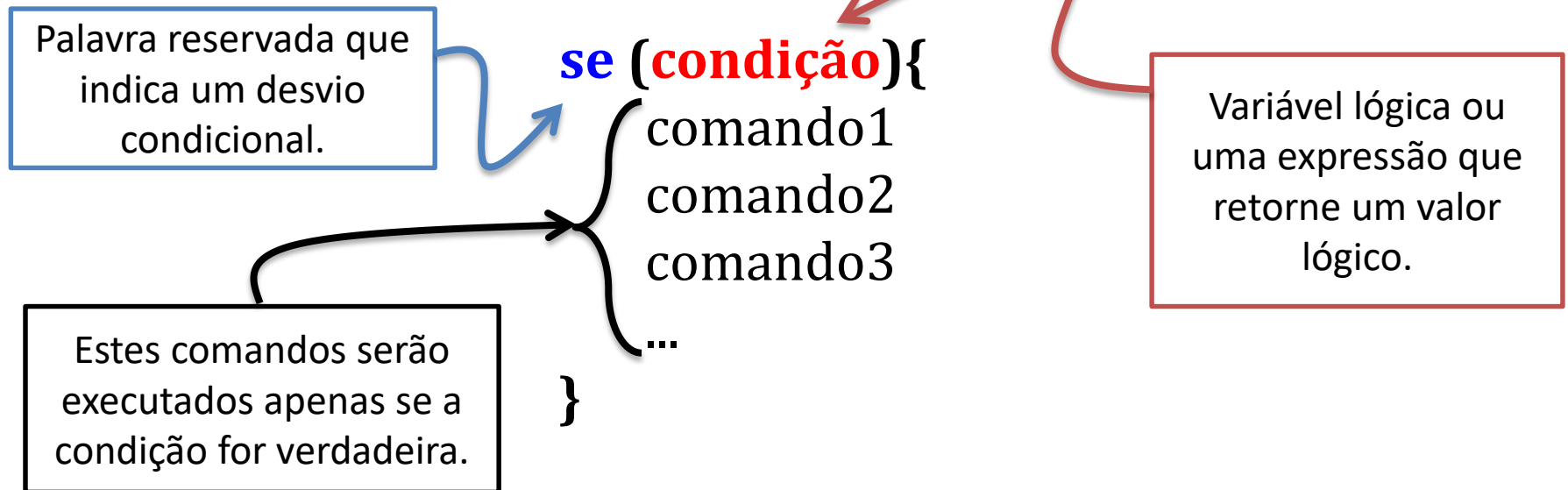
```
se (condição) {  
    comando1  
    comando2  
    comando3  
    ...  
}
```

Variável lógica ou uma expressão que retorne um valor lógico.

Estruturas de decisão

- **Comando se**

- Em Portugal, se desejarmos que um determinado conjunto de comandos seja executado **apenas** se uma condição for satisfeita, devemos usar a seguinte sintaxe:



Estruturas de decisão

- **Comando se**
 - Por exemplo, veja a seguinte função:

```
funcao logico ehMaiorDeIdade(inteiro idade){  
    logico teste = falso  
    se(idade > 18){  
        teste = verdadeiro  
    }  
    retorne teste  
}
```

- O que ela retorna se a chamarmos assim?

```
ehMaiorDeIdade(20)
```

Estruturas de decisão

- Comando se
 - Por exemplo, veja a seguinte função:

```
funcao logico ehMaiorDeIdade(inteiro idade){  
    logico teste = falso  
    se(idade > 18){  
        teste = verdadeiro  
    }  
    retorne teste  
}
```

Esta linha só
será executada
se a idade for
maior que 18.

- O que ela retorna se a chamarmos assim?

ehMaiorDeIdade(20)

verdadeiro

Estruturas de decisão

- Comando se

- Outro exemplo:

```
funcao logico ehVogal(caracter c){  
    logico resultado = falso  
    se(c == 'A' ou c == 'a' ou  
       c == 'E' ou c == 'e' ou  
       c == 'I' ou c == 'i' ou  
       c == 'O' ou c == 'o' ou  
       c == 'U' ou c == 'u'){  
        resultado = verdadeiro  
    }  
    retorne resultado  
}
```

- O que ela retorna se a chamarmos assim?

ehVogal('0')

Estruturas de decisão

- Comando se

- Outro exemplo:

Esta linha só
será executada
se c for uma
vogal.

```
funcao logico ehVogal(caracter c){  
    logico resultado = falso  
    se(c == 'A' ou c == 'a' ou  
       c == 'E' ou c == 'e' ou  
       c == 'I' ou c == 'i' ou  
       c == 'O' ou c == 'o' ou  
       c == 'U' ou c == 'u'){  
        resultado = verdadeiro  
    }  
    retorne resultado  
}
```

- O que ela retorna se a chamarmos assim?

ehVogal('0')

falso

Estruturas de decisão

- **Exemplos**

1. Escreva uma função que recebe dois números inteiros **distintos** e retorna o maior.

2. Deseja-se calcular a conta de consumo de energia elétrica de um consumidor. Para isso, escreva uma **função** que recebe o **preço** do KW, a **quantidade** de KW consumido e o **valor mínimo** a se pagar. Ao final, a função **retorna** o **total a pagar**:

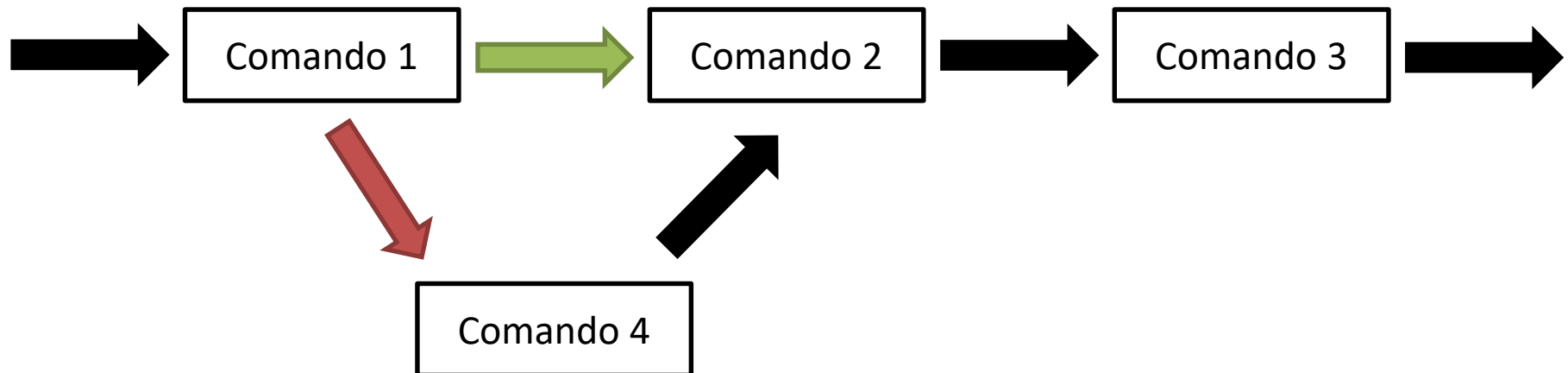
- Total a pagar = preço x quantidade
- Restrição: existe um valor mínimo a ser pago.

Escreva um **programa** para **ler** o preço do KW, a quantidade de KW consumido e o valor mínimo a se pagar. **Chame a função** recém construída e **escreva** o total a pagar.

Estruturas de decisão

- **Continuando**

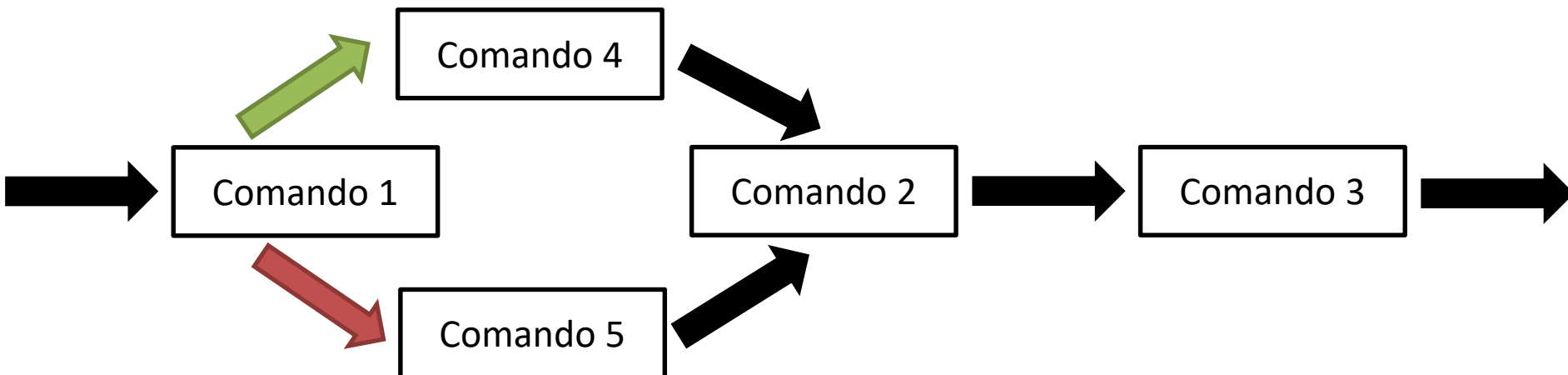
- Com isso, já sabemos desviar o fluxo de execução do programa para executar um **bloco de comandos alternativo** se uma condição for satisfeita.



Estruturas de decisão

- **Continuando**

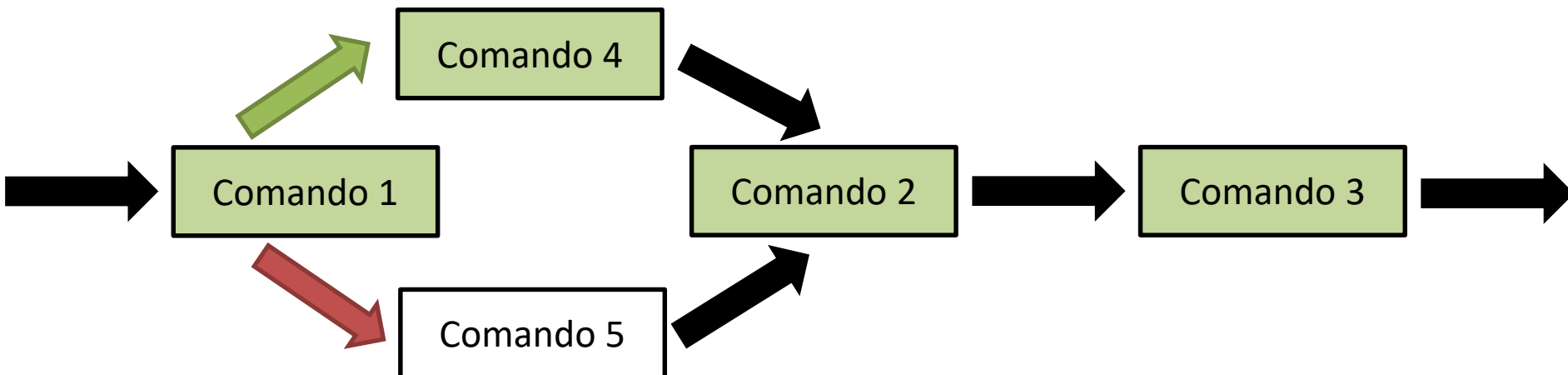
- Porém, algumas vezes, precisamos criar uma **bifurcação** no processamento.
- Dependendo da condição, **1** dentre **2 possíveis blocos** de comandos será executado.



Estruturas de decisão

- **Continuando**

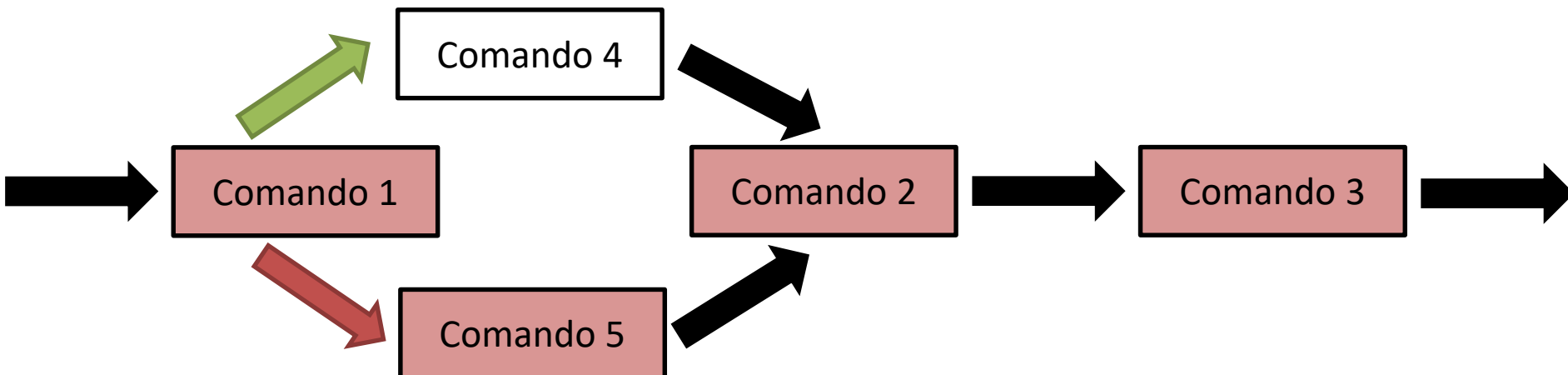
- Porém, algumas vezes, precisamos criar uma **bifurcação** no processamento.
- Dependendo da condição, 1 dentre **2 possíveis blocos** de comandos será executado.



Estruturas de decisão

- **Continuando**

- Porém, algumas vezes, precisamos criar uma **bifurcação** no processamento.
- Dependendo da condição, 1 dentre **2 possíveis blocos** de comandos será executado.



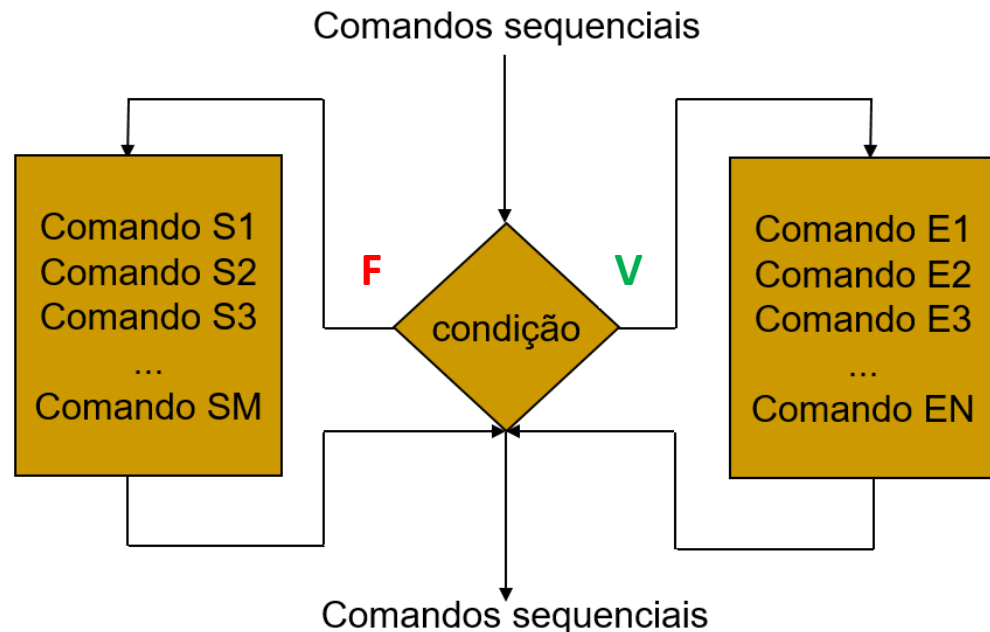
Estruturas de decisão

- **Comando se, senão**

- Nestes casos, sempre haverá **desvio de processamento**.
- Restará apenas decidir por qual caminho a execução seguirá.

- Por exemplo:

➤ Se a senha inserida estiver correta, o login deve ser feito, caso contrário, uma mensagem de erro deve ser gerada.



Estruturas de decisão

- **Comando se, senão**

- Em Portugal, a sintaxe da estrutura que realiza este tipo de desvio é a seguinte:


```
se (condição){  
    comando1  
    comando2  
    comando3  
    ...  
} senao {  
    comando4  
    comando5  
    comando6  
    ...  
}
```

Estruturas de decisão

- **Comando se, senão**

- Em Portugal, a sintaxe da estrutura que realiza este tipo de desvio é a seguinte:

Se a condição for verdadeira, este bloco será executado.

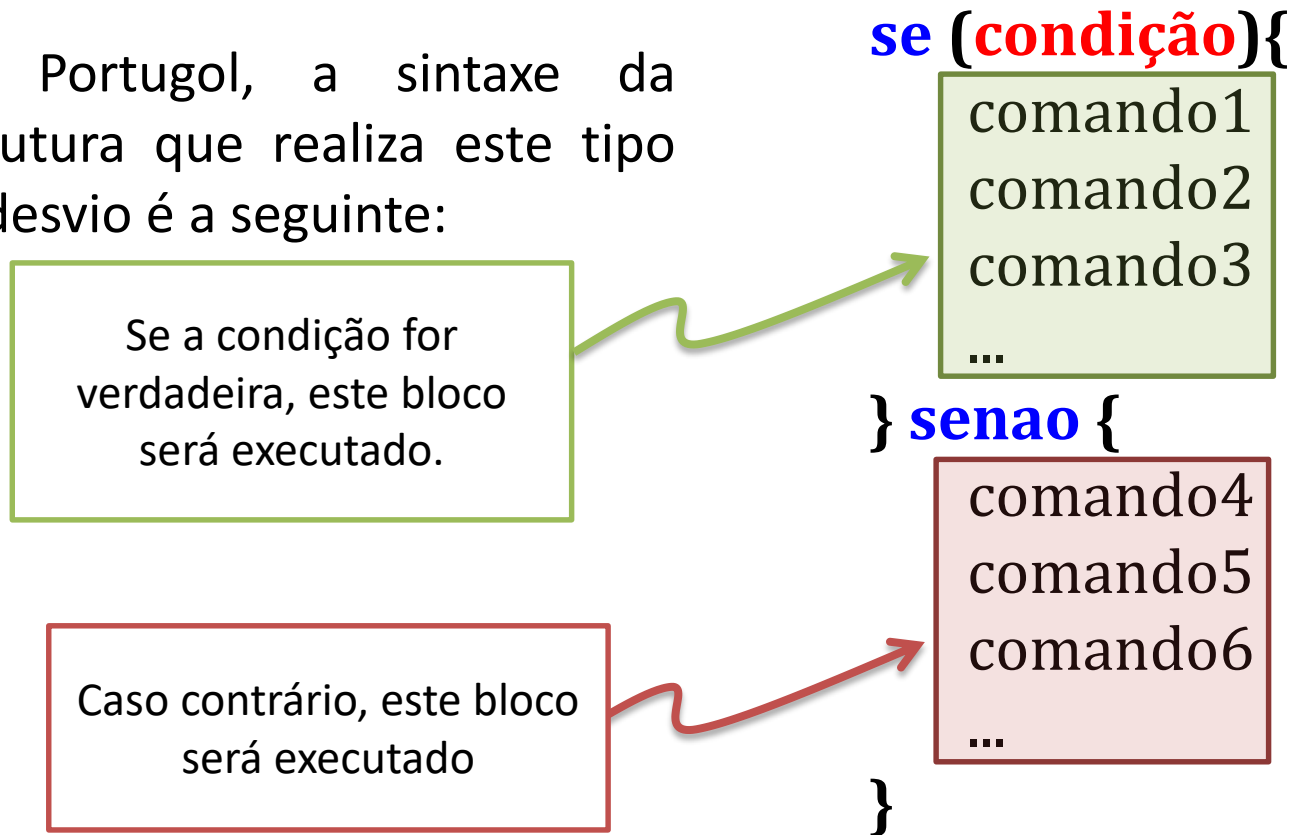


```
se (condição) {  
    comando1  
    comando2  
    comando3  
    ...  
} senao {  
    comando4  
    comando5  
    comando6  
    ...  
}
```

Estruturas de decisão

- **Comando se, senão**

- Em Portugal, a sintaxe da estrutura que realiza este tipo de desvio é a seguinte:



Estruturas de decisão

- **Comando se, senão**

- Por exemplo, usando a função “ehMaiorDeIdade”, criada anteriormente:

```
funcao inicio(){  
    inteiro idade  
  
    leia(idade)  
  
    se(ehMaiorDeIdade(idade)){  
        escreva("Maior de idade")  
    } senao {  
        escreva("Menor de idade")  
    }  
}
```

Estruturas de decisão

- **Exemplos**

3. Usando o comando “se, senão”, escreva uma **função** para retornar o maior dentre dois números. Teste sua função em um **módulo início**.

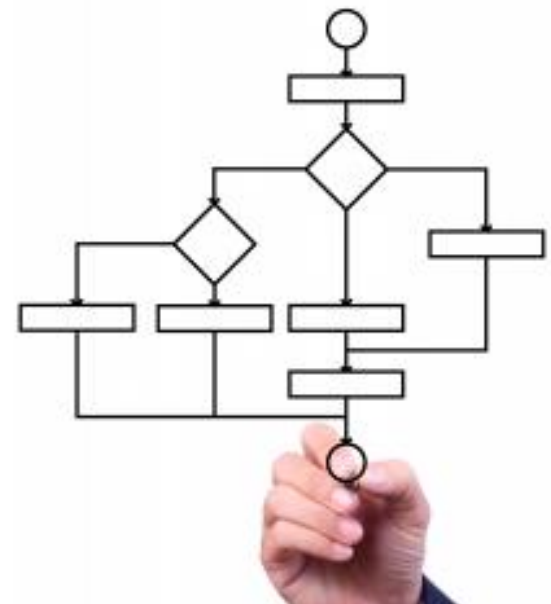
4. Usando o comando “se, senão”, escreva uma **função** para retornar o menor dentre dois números. Teste sua função em um **módulo início**.

5. Use as funções dos exemplos 3 e 4 e escreva um **programa** para **ler** dois números e **exibi-los** em ordem crescente.

6. Escreva uma **função** para determinar se um número inteiro é par ou ímpar. Ela deve **retornar** a cadeia “par” ou “ímpar”, dependendo do caso. Teste sua função em um módulo início.

Estruturas de decisão

- **Aninhamentos de comandos se**
 - Muitas vezes, a implementação de um algoritmo exige a utilização de muitos comandos **se**.
 - Os comandos **se**, inclusive, podem ser utilizados **dentro** de outros comandos **se**.
 - Nestes casos, é criado um **aninhamento** de comandos **se**.
 - Se este aninhamento não for construído corretamente, pode gerar muitos problemas no seu programa.



Estruturas de decisão

- **Aninhamentos de comandos se**
 - Por exemplo:
 - A **situação** de um aluno pode ser obtida por um programa que segue as seguintes regras:
 - $\text{media} \geq 7 \rightarrow$ aluno aprovado
 - media entre 3 e 7 \rightarrow prova final
 - $\text{media} < 3 \rightarrow$ aluno reprovado



Estruturas de decisão

- Aninhamentos de comandos se
 - Por exemplo:


```
funcao inicio(){  
    real media  
  
    leia(media)  
  
    se(media >= 7.0){  
        escreva("Aprovado")  
    } senao se (media >= 3) {  
        escreva("Prova Final")  
    } senao {  
        escreva("Reprovado")  
    }  
}
```

Estruturas de decisão

- **Aninhamentos de comandos se**
 - Por exemplo:

```
funcao inicio(){  
    real media  
  
    leia(media)  
  
    se(media >= 7.0){  
        escreva("Aprovado")  
    } senao se (media >= 3) {  
        escreva("Prova Final")  
    } senao {  
        escreva("Reprovado")  
    }  
}
```

Se a media for maior ou igual a 7, este comando será executado.



Estruturas de decisão

- **Aninhamentos de comandos se**

- Por exemplo:

Caso contrário, se for pelo menos maior ou igual a 3, este comando será executado.

```
funcao inicio(){  
    real media  
  
    leia(media)  
  
    se(media >= 7.0){  
        escreva("Aprovado")  
    } senao se (media >= 3) {  
        escreva("Prova Final")  
    } senao {  
        escreva("Reprovado")  
    }  
}
```

Se a media for maior ou igual a 7, este comando será executado.

Estruturas de decisão

- **Aninhamentos de comandos se**
 - Por exemplo:

Caso contrário, se for pelo menos maior ou igual a 3, este comando será executado.

```
funcao inicio(){  
    real media  
  
    leia(media)  
  
    se(media >= 7.0){  
        escreva("Aprovado")  
    } senao se (media >= 3) {  
        escreva("Prova Final")  
    } senao {  
        escreva("Reprovado")  
    }  
}
```

Se a media for maior ou igual a 7, este comando será executado.

Se não se enquadrar nas alternativas anteriores, este comando será executado.

Estruturas de decisão

- Aninhamentos de comandos se
 - Por exemplo:

```
funcao inicio(){  
    real media  
  
    leia(media)  
  
    se(media >= 7.0){  
        escreva("Aprovado")  
    } senao se(media >= 3) {  
        escreva("Prova Final")  
    } senao {  
        escreva("Reprovado")  
    }  
}
```



É muito importante deixar sempre o código bem organizado e indentado. Com isso, você não se perde com relação aos “se” e os “senão”

Estruturas de decisão

- **Exemplo**

7. Observe o trecho de programa ao lado, considerando L1, L2 e L3 como **variáveis lógicas**.

- O que será escrito pelo programa se, para L1, L2 e L3, forem lidos, respectivamente:

- a) V, V e V
- b) F, V e V
- c) F, V e F

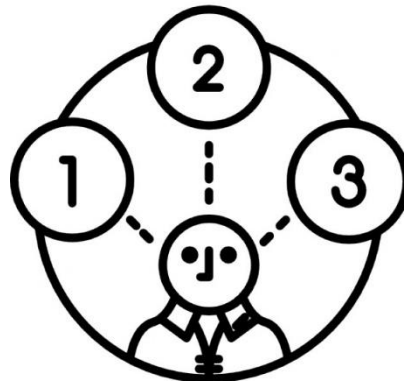
- Quais valores devem ser lidos para L1, L2 e L3, para que seja escrito apenas “E”?

```
se(L1){  
    escreva("A")  
} senao se(L2) {  
    se(L3){  
        escreva("B")  
    } senao {  
        escreva("C")  
        escreva("D")  
    }  
} senao {  
    escreva("E")  
}
```

Estruturas de decisão

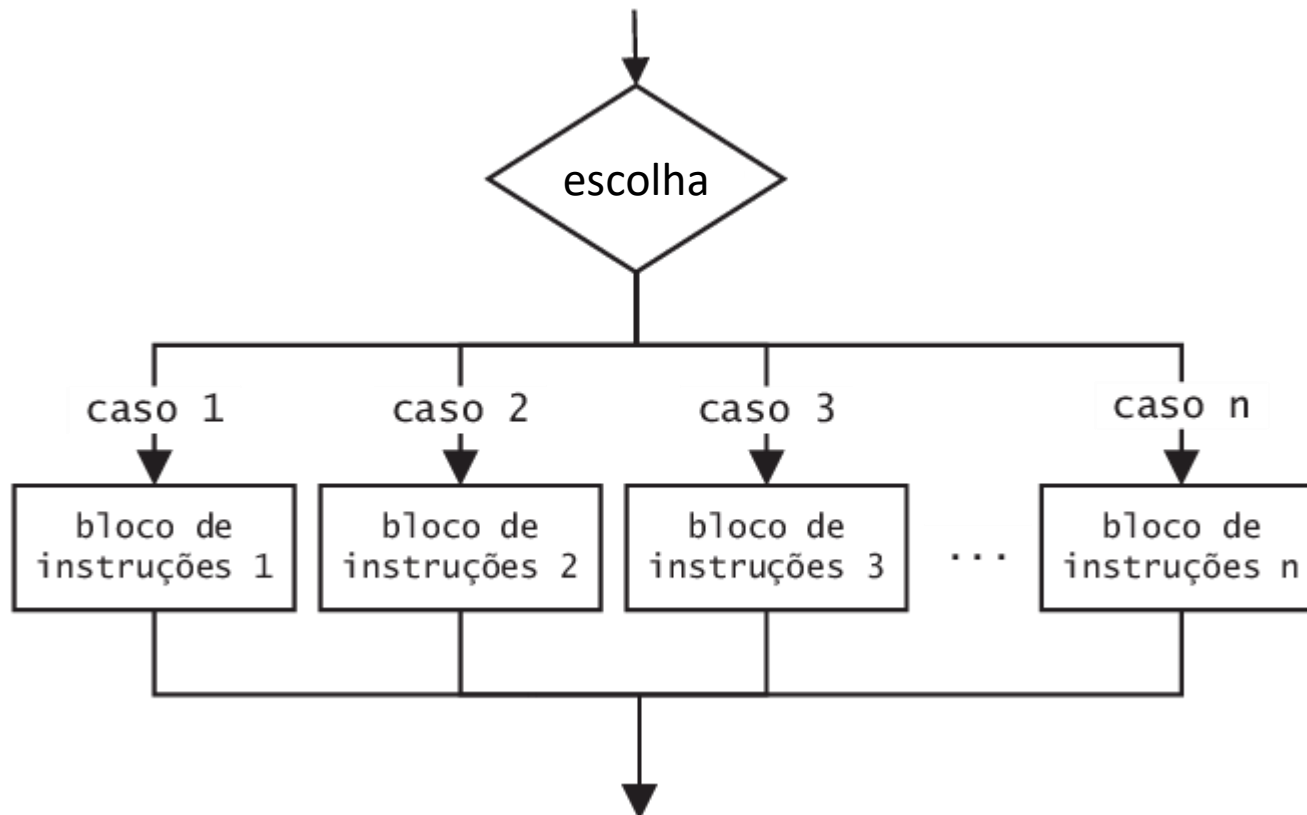
- **Escolha-caso**

- Existe ainda uma outra maneira de implementar um desvio condicional.
- Algumas vezes, sabemos uma lista de valores que uma variável poderá assumir e as ações que devem ser tomadas em cada caso.
- Nesta situação, podemos usar o **escolha-caso**.



Estruturas de decisão

- Escolha-caso



Estruturas de decisão

- Escolha-caso

```
inteiro opcao

escreva("1) Elogio \n")
escreva("2) Ofensa \n")
escreva("3) Sair \n\n")

escreva("Escolha uma opção: ")
leia(opcao)

escolha (opcao){
  caso 1:
    escreva ("Você é lindo(a)!")
    pare // Impede que as instruções do caso 2 sejam executadas
  caso 2:
    escreva ("Você é um monstro!")
    pare // Impede que as instruções do caso 2 sejam executadas
  caso 3:
    escreva ("Tchau!")
    pare
  caso contrario: // Será executado para qualquer opção diferente de 1, 2 ou 3
    escreva ("Opção Inválida !")
}
```

Estruturas de decisão

- Escolha-caso

```
inteiro opcao
```

```
escreva("1) Elogio \n")  
escreva("2) Ofensa \n")  
escreva("3) Sair \n\n")
```

```
escreva("Escolha uma opção: ")  
leia(opcao)
```

```
escolha opcao
```

```
caso 1:
```

```
    escreva ("Você é lindo(a)!")
```

```
    pare // Impede que as instruções do caso 2 sejam executadas
```

```
caso 2:
```

```
    escreva ("Você é um monstro!")
```

```
    pare // Impede que as instruções do caso 2 sejam executadas
```

```
caso 3:
```

```
    escreva ("Tchau!")
```

```
    pare
```

```
caso contrario: // Será executado para qualquer opção diferente de 1, 2 ou 3
```

```
    escreva ("Opção Inválida !")
```

```
}
```



Atenção! A variável que irá entrar no comando “escolha” só pode ser dos tipos **inteiro** ou **caracter**.

Estruturas de decisão

- **Exercícios**

1 - Escreva uma **função** que **recebe três números inteiros** e **retorna o menor deles**. Teste sua função em um **módulo início**.

2 - Escreva um **programa** dividido em **subprogramas** para determinar o grau de obesidade de uma pessoa, sendo fornecidos o peso e a altura da pessoa. O grau de obesidade é determinado pelo índice da massa corpórea ($\text{Massa} = \text{Peso} / \text{Altura}^2$) através da tabela abaixo:

MASSA CORPÓREA	GRAU DE OBESIDADE
< 26	Normal
≥ 26 e < 30	Obeso
≥ 30	Obeso Mórbido

• Exercícios

3. Escreva um **programa** que lê um número inteiro e imprime se ele é positivo, negativo ou nulo.

4. Escreva um **programa** que **leia** a sigla do estado em que uma pessoa nasceu e **escreva** uma das mensagens: “carioca”, “paulista”, “mineiro” ou “outro estado” de acordo com estado informado.

5. Escreva uma **função** que **recebe** uma data (dia, mês e ano em separado) e **retorne** um valor lógico indicando se a data é válida ou não.

6. Escreva uma função que receba um caractere como parâmetro e retorne verdadeiro caso o caractere seja uma consoante, e falso caso contrário.

7. Segundo uma tabela médica, o peso ideal está relacionado com a altura e o sexo. Escreva uma função que receba a altura e o sexo de uma pessoa, calcule e imprima o seu peso ideal, utilizando as seguintes fórmulas:

➤ Para homens: $72.7 \times h - 58$

➤ Para mulheres: $63.1 \times h - 44.7$

• Exercícios

8. A prefeitura do Rio de Janeiro abriu uma linha de crédito para os funcionários estatutários. O valor máximo da prestação não poderá ultrapassar 30% do salário bruto. Escreva uma **função** que **recebe** o salário bruto e o valor da prestação e **retorna** um valor lógico indicando se o empréstimo pode ou não ser concedido.

9. Um estacionamento cobra de seus clientes o valor de R\$ 2,00 por hora ou fração. Se um cliente ficar mais de 4 horas, é cobrado o valor total de uma diária, igual a R\$ 10,00. Crie uma função que recebe o tempo que o cliente passou no estacionamento em minutos e retorna o valor total a ser pago.

10. Um comerciante comprou um produto e quer vendê-lo com um lucro de 45% se o valor do produto for menor que R\$ 20,00; caso contrário o lucro será de 30%. Escreva uma função que receba o valor do produto e imprima o valor da venda.

Estruturas de decisão

- **Exercícios**

11. Um clube de futebol deseja aumentar o salário de seus jogadores. O reajuste deve obedecer a seguinte tabela:

SALÁRIO ATUAL (R\$)	AUMENTO
0,00 a 1.000,00	20%
1.000,01 a 5.000,00	10%
Acima de 5.000,00	0%

Escreva um **programa** para ler o nome e o salário atual de um jogador. Depois, o programa deve escrever o nome, o salário atual e o salário reajustado. Faça uma **função** para calcular o reajuste do salário. A função deve **receber** o valor do salário atual e **retornar** o salário reajustado.

- **Exercícios**

12. Faça um programa para calcular a conta final de um hóspede de um hotel, considerando que:

- Serão lidos o nome do hóspede, o tipo do apartamento utilizado (A, B, C ou D), o número de diárias utilizadas pelo hóspede e o valor do consumo interno do hóspede.
- O valor da diária é determinado pela seguinte tabela:

TIPO DO APTO.	VALOR DA DIÁRIA (R\$)
A	150,00
B	100,00
C	75,00
D	50,00

- O valor total das diárias é o número de diárias usadas multiplicado pelo valor da diária.
- O subtotal é a soma do valor total das diárias e o valor do consumo interno.
- O valor da taxa de serviço equivale a 10% do subtotal.
- O total geral resulta da soma do subtotal com a taxa de serviço.

O programa deve imprimir a conta final, contendo: o nome do hóspede, o tipo do apartamento, o número de diárias utilizadas, o valor unitário da diária, o valor total das diárias, o valor do consumo interno, o subtotal, o valor da taxa de serviço e o total geral. (Obs: Procure **dividir** ao máximo o programa em **subprogramas**)

• Exercícios

13. Escreva uma função chamada “pegaDigito” que recebe um número inteiro e retorna o dígito em uma posição desejada. Considere o dígito das unidades como posição 0, o dígito das dezenas como posição 1, e assim por diante. Se o número não possuir um dígito na posição indicada, a função deve retornar -1 e escrever na tela “Erro: posição inválida”.

Exemplos:

- pegaDigito(41253, 3) -> retorna 1
- pegaDigito(41253, 0) -> retorna 3
- pegaDigito(41253, 5) -> retorna -1 e escreve “Erro: posição inválida”.

Dica: pense em utilizar as operações de divisão, resto da divisão e potência.

14. Um número é **estranho** se tem no máximo 5 dígitos e todos eles são menores do que o seu dígito menos significativo. Escreva uma função que retorna um valor lógico indicando se um número é **estranho** .

- Exercícios

15. Um número é **horário** se estiver representando a indicação de um relógio digital. Escreva uma função que retorna um valor lógico indicando se um número inteiro é **horário**. Ex: 11:25h = 1125.

16. Um número é **pequeno** se é um múltiplo de 3 e, além disso, o seu dígito menos significativo é ímpar. Escreva uma função que retorna um valor lógico indicando se um número inteiro é **pequeno**.

17. Um número é **assimétrico** se, além de ser **pequeno**, for maior do que 10 e menor do que 1000. Escreva uma função que retorna um valor lógico indicando se um número é **assimétrico**.

18. Um número é **analfabeto** se possui no máximo 4 dígitos e o total de dígitos pares for igual ao total de dígitos ímpares. Escreva uma função que retorna um valor lógico indicando se um número é analfabeto.

19. Um número é **louco** se for **horário** e **analfabeto**. Escreva uma função que retorna um valor lógico indicando se um número é louco.

Estruturas de decisão

FIM