

## **Operadores Relacionais e Lógicos / Estrutura Condicional**

### **1. Operadores Relacionais e Lógicos**

#### **1.1. Operadores Relacionais**

Os operadores relacionais são usados para realizar comparações entre dois valores de mesmo tipo primitivo. Tais valores são representados por constantes, variáveis ou expressões aritméticas.

**Exemplos:**

<b>Símbolo</b>	<b>Nome do Operador</b>	<b>Exemplo</b>	<b>Significado</b>
<b>&gt;</b>	Maior que	$x > y$	x é maior que y?
<b>&gt;=</b>	Maior ou igual	$x >= y$	x é maior ou igual a y ?
<b>&lt;</b>	Menor que	$x < y$	x é menor que y?
<b>&lt;=</b>	Menor ou igual	$x <= y$	x é menor ou igual a y ?
<b>==</b>	Igualdade	$x == y$	x é igual a y?
<b>!=</b>	Diferente de	$x != y$	x é diferente de y?

A seguir, são apresentados alguns exemplos de comparações válidas:

<b>Comparação Válida</b>	<b>Exemplo</b>	<b>Observações</b>
variável e constante	$X == 3$	Comparação se o conteúdo da variável <b>X</b> é <b>igual a</b> constante <b>3</b> .
variável e variável	$A != B$	Comparação se o conteúdo da variável <b>A</b> é <b>diferente</b> do conteúdo da variável <b>B</b> .
variável e expressão	$Y == W + J$	Comparação se o conteúdo da variável <b>Y</b> é igual ao valor da expressão <b>W + J</b> .
expressão e expressão	$X + 1 < Y + 4$	Comparação se o valor da expressão <b>X + 1</b> é menor que o valor da expressão <b>Y + 4</b> .

**Exemplos:**

a)  $2 * 4 == 24 / 3$   
 $8 == 8$

**Resultado: V** (Verdadeiro)

b)  $15 \% 4 == 19 \% 6$

$$3 == 1$$

**Resultado: F** (Falso)

## 1.2. Operadores Lógicos

Os operadores lógicos permitem que mais de uma condição seja testada em uma única expressão, ou seja, pode-se fazer mais de uma comparação (teste) ao mesmo tempo. A Tabela abaixo apresenta os operadores lógicos que utilizaremos nesta disciplina.

Operação	Operador	Operador em Portugol Studio
Negação	<b>NÃO</b>	<b>nao</b>
Conjunção	<b>E</b>	<b>e</b>
Disjunção (não-exclusiva)	<b>OU</b>	<b>ou</b>

**E**, **OU** e **NÃO** são os principais operadores lógicos, base para a construção de sistemas digitais e da Lógica proposicional. Os operadores **E** e **OU** são operadores binários, ou seja, necessitam de dois elementos, enquanto o **NÃO** é unário. Na computação, esses elementos são normalmente variáveis binários, cujos possíveis valores atribuídos são 0 ou 1. Porém, a lógica empregada para essas variáveis serve também para sentenças (frases) da linguagem humana, onde se esta for verdade corresponde ao valor 1, e se for falsa corresponde ao valor 0. Essas frases são chamadas proposições. Uma **proposição** é uma **afirmação** que pode ser dita como verdadeira ou falsa.

### Exemplos de proposição:

- Está chovendo.
- Vai ter aula hoje.
- O animal é um cão.

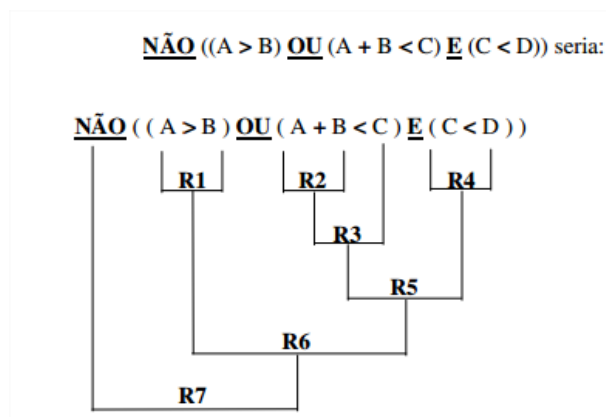
**PRIORIDADE ENTRE OS OPERADORES LÓGICOS:** A ordem de prioridade dos operadores obedece a ordem do quadro abaixo. Ou seja, numa expressão que tenha os operadores **NÃO**, **E** e **OU**, primeiro resolve-se o **NÃO**, depois o **E** e por último o **OU**.

Operadores	Ordem de prioridade
<b>nao</b>	1ª
<b>e</b>	2ª
<b>ou</b>	3ª

Esta ordem de prioridade só pode ser quebrada com o uso de parênteses. Com relação aos outros tipos de operadores, a ordem de prioridade ficará da seguinte forma:

Operadores	Ordem de prioridade
<b>nao</b>	1ª
<b>potencia()</b>	2ª
<b>*, /, e</b>	3ª
<b>+, -, ou</b>	4ª
<b>&lt;, &gt;, ==, ...</b>	5ª

Assim, a prioridade de cálculo da expressão:



Note que, apesar do operador **nao** ter a maior prioridade de cálculo, o mesmo foi aplicado por último devido ao uso de parênteses.

### 1.3. TABELA-VERDADE

- **E (e)** - O operador **e** analisa dois valores lógicos, retornando verdadeiro **apenas** quando **ambos** são **verdadeiros**.

A tabela-verdade para o **e** é a seguinte:

A	B	A e B
F	F	F
V	F	F
F	V	F
V	V	V

Analogia – imagine as seguintes proposições:

- 1) A: tem quatro patas;
- 2) B: mia;
- 3) C: é um gato.

Podemos então dizer que a proposição **C** é verdadeira se as proposições **A e B** forem verdadeiras. Se tem quatro patas mas não mia, ou mia mas não tem quatro patas, não podemos dizer que é um gato. Portanto, podemos dizer que “**C == A e B**”.

- **Ou (ou)**: o operador **ou** analisa dois valores lógicos, retornado falso apenas quando ambos são falsos. Ex.

A tabela-verdade para o **ou** é a seguinte:

A	B	A ou B
F	F	F
V	F	V
F	V	V
V	V	V

Analogia – imagine que Samuel torce para o time A e não gosta do Time B. Agora imagine as seguintes proposições:

1. A: Time A é campeão brasileiro;
2. B: Time B é rebaixado;
3. C: Samuel feliz.

Samuel ficaria feliz se A ou B fossem proposições verdadeiras. Ficaria ainda muito feliz se ambas fossem verdadeiras. Logo, “**C == A ou B**”. Assim não é necessário que A e B sejam ambas verdadeiras para que C o seja.

- **Não (nao)**: o operador **nao** é um operador unário (ao contrário dos anteriores, que são binários), o que significa que ele age sobre um único valor. O **nao** retorna o contrário do valor analisado.

A tabela-verdade para o **nao** é a seguinte:

A	nao A
V	F
F	V

### Exemplos:

Expressão	Quando eu não saio?
Se chover <i>e</i> relampejar, eu não saio.	Somente quando chover e relampejar ao mesmo tempo (apenas 1 possibilidade).
Se chover <i>ou</i> relampejar, eu não saio.	Somente quando chover, somente quando relampejar ou quando chover e relampejar ao mesmo tempo (3 possibilidades).

A) Se (**salario > 500 e salario < 1800**){

Escreva (“Salário válido para financiamento”)

}

**Senao**{

Escreva (“Salário fora da faixa permitida para financiamento”)

}

B) Se (**idade < 18 ou idade > 95**) {

Escreva (“Você não pode fazer carteira de motorista”)

}

**Senao**{

Escreva (“Você pode possuir carteira de motorista”)

}

C) Se (**idade >= 18 e idade <= 95 e aprovado\_exame = “sim”**){

Escreva(“Sua carteira de motorista estará pronta em uma semana”)

}

**Senao**{

Escreva(“Você não possui idade permitida *ou* não passou nos testes”)

}

### Exercícios

1) Com base na tabela-verdade, determine o resultado lógico das expressões mencionadas (Verdadeira ou Falsa). Considere para as respostas os seguintes valores: X=1, A=3, B=5, C=8 e D=7.

a) não ( $X > 3$ )

f) ( $X \geq 2$ )

b) ( $X < 1$ ) e ( $\text{nao}(B > D)$ )

g) ( $X < 1$ ) e ( $B \geq D$ )

c)  $\text{nao}(D < 0)$  e ( $C > 5$ )

h) ( $D < 0$ ) ou ( $C > 5$ )

d)  $\text{nao}((X > 3) \text{ ou } (C < 7))$

i)  $\text{nao}(D > 3)$  ou  $\text{nao}(B < 7)$

e) ( $A > B$ ) ou ( $C > B$ )

j) ( $A > B$ ) ou  $\text{nao}(C > B)$

2) Para A = V, B = V e C = F, qual o resultado da avaliação das seguintes expressões:

a) (A e B) ou (A ou B)

b) (A ou B) e (A e C)

c) A ou C e B ou A e  $\text{nao B}$

## 2. Estrutura Condicional Simples

### 2.1. Introdução

Existem problemas que podem ter mais de um caminho a ser seguido para seleção correta, ou existem restrições em suas soluções. Assim, quem for executar o algoritmo, num determinado momento, deverá tomar a decisão de qual caminho seguir para chegar à solução correta do problema, com base nas restrições impostas.

Esta decisão, do caminho a ser seguido, tem de estar fundamentada em alguma lógica. E é função de quem escreve o algoritmo, dar condições para o sujeito que o executa, fazer a escolha correta do caminho para chegar à solução do problema.

Um exemplo simples de problema que pode ter metodologias diferentes em sua solução, são os cálculos das raízes de equações polinomiais do 2º grau ( $ax^2 + bx + c = 0$ ). Se o termo  $b^2 - 4ac$  for maior ou igual a zero, as raízes serão reais, caso contrário, serão complexas.

Não seria correto escrever um algoritmo que resolvesse somente a parte de raízes reais, ou, a parte complexa, pois a metodologia que vai ser usada somente será conhecida em tempo de execução do algoritmo. Logo, temos que descrever a solução para ambas as metodologias, mas deixando claro para quem vai executá-la, que ele terá de fazer uma escolha entre os caminhos possíveis.

Mas se deixássemos a critério de quem executa o algoritmo, será que ele sempre escolheria o caminho correto? Por isso, é que, além de dizermos que existe mais de um caminho para a solução do problema, ainda temos de prover subsídios para que o mesmo faça a escolha do caminho correto. Assim, o simples fato de informar que a solução do problema tem duas ou mais metodologias diferentes em sua solução, não funciona. Temos ainda, que dizer em quais situações ele utilizará a metodologia A, e em que outras ele utilizará a B. No caso das raízes de polinômio do 2º grau, poderíamos utilizar a seguinte descrição narrativa:

**se** o resultado de  $b^2 - 4ac$  for maior ou igual a zero,  
    utilize a metodologia de cálculo de raízes reais,  
**senão**  
    utilize a metodologia de cálculo de raízes complexas.

### 2.2. Comando SE

Aqui veremos como dizer a um algoritmo quando um conjunto de instruções deve ser executado. Esta determinação é estabelecida se uma condição for verdadeira. Mas o que seria esta condição? Ao executar um teste lógico teremos como resultado um valor verdadeiro ou falso. A condição descrita anteriormente nada mais é que um teste lógico.

Se este teste lógico resultar verdadeiro, as instruções definidas dentro do desvio condicional serão executadas. Se o teste for falso, o algoritmo pulará o trecho e continuará sua execução a partir do ponto onde o desvio condicional foi finalizado.

O desvio condicional que foi acima apresentado é considerado simples e conhecido como o comando **se**.

```
logico condicao = verdadeiro
se (condicao)
{
    //Instruções a serem executadas se o desvio for verdadeiro
}

inteiro x = 5
se (x > 3)
{
    //Instruções a serem executadas se o desvio for verdadeiro
}
```

A sintaxe é respectivamente a palavra reservada **se**, a condição a ser testada entre parênteses e as instruções que devem ser executadas entre chaves caso o desvio seja verdadeiro.

A figura à esquerda ilustra um algoritmo que verifica se o número digitado pelo usuário é zero. Ele faz isso usando um desvio condicional. Note que se o teste for verdadeiro exibirá uma mensagem, no caso falso nenhuma ação é realizada.

O exemplo à direita ilustra em Portugol o mesmo algoritmo do fluxograma.

### 2.3. Se-senao

Agora vamos imaginar que se a condição for falsa, outro conjunto de comandos deve ser executado. Quando iremos encontrar esta situação?

Imagine um programa onde um aluno com média final igual ou maior a 60 é aprovado. Se quisermos construir um algoritmo que depois de calculada a média, seja mostrada na tela uma mensagem indicando se o aluno foi aprovado ou reprovado. Como fazer isto?

Utilizando o comando **se** junto com o **senao**.

Sua sintaxe é simples, basta no termino do comando **se** ao lado do fechamento de chaves, colocar o comando **senao** e entre chaves colocar as instruções a serem executadas caso o comando **se** for **falso**.

```
logico condicao = falso
se (condicao)
{
    //Instruções a serem executadas se o desvio for verdadeiro
}
senao
{
    //Instruções a serem executadas se o desvio for falso
}
```

O fluxograma abaixo ilustra um algoritmo que tem como entrada a hora, e ao analisá-la, deverá informar se é dia ou noite.



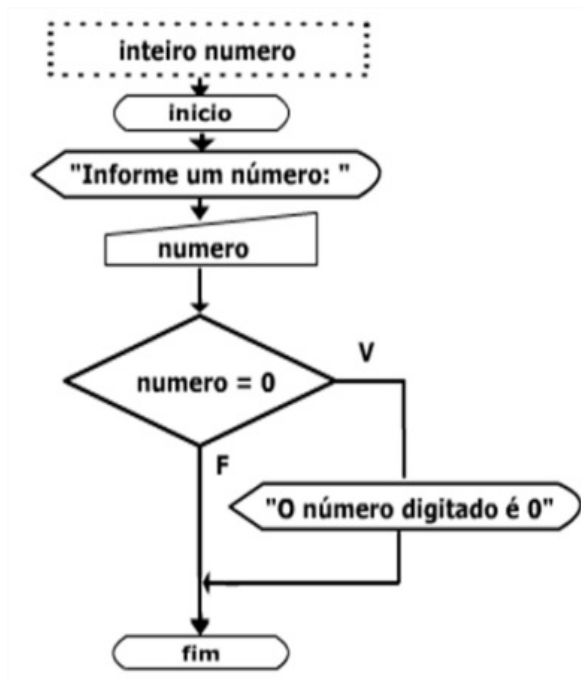
```
programa
{
    funcao inicio()
    {

        inteiro hora

        escreva ("Digite a hora: ")
        leia (hora)

        se (hora >= 6 e hora <= 18)
        {
            escreva ("É dia")
        }
        senao
        {
            escreva ("É noite")
        }
    }
}
```

O exemplo a seguir ilustra em portugol o mesmo algoritmo do fluxograma abaixo:



```

programa
{
  funcao inicio()
  {

    inteiro num

    escreva ("Digite um número: ")
    leia (num)

    se (num==0)
    {
      escreva ("O número digitado é 0")
    }

  }
}
  
```

## 2.4. Se-senao se

Imagine que você precise verificar a nota da prova de um aluno e falar se ele foi muito bem, bem, razoável ou mau em uma prova. Como fazer isto?

Quando você precisa verificar se uma condição é verdadeira, e se não for, precisa verificar se outra condição é verdadeira, uma das formas de se fazer esta verificação é utilizando do comando **se...senao se**. A sua sintaxe é parecida com a do **senao**, mas usando o comando **se** imediatamente após escrever o comando **senao**.

```

logico condicao = falso
logico condicao2 = verdadeiro
se (condicao)
{
  //Instruções a serem executadas se o desvio for verdadeiro
}
senao se (condicao2)
{
  //Instruções a serem executadas se o desvio anterior for falso e este desvio for verdadeiro
}
  
```

Pode-se também colocar o comando **senao** no final do último **senao se**, assim quando todos os testes falharem, ele irá executar as instruções dentro do **senão**.

```

se (12 < 5)
{
  //Instruções a serem executadas se o desvio for verdadeiro
}
senao se ("palavra" == "texto")
{
  //Instruções a serem executadas se o desvio anterior for falso e este desvio for verdadeiro
}
senao
{
  //Instruções a serem executadas se o desvio anterior for falso
}
  
```

O exemplo a seguir ilustra a resolução do em Portugol de avisar se o aluno foi muito bem, bem, razoável ou mau em uma prova.

```

programa
{
    funcao inicio()
    {
        real nota
        leia(nota)
        se(nota >= 9)
        {
            escreva("O aluno foi um desempenho muito bom na prova")
        }
        senao se (nota >= 7)
        {
            escreva("O aluno teve um desempenho bom na prova")
        }
        senao se (nota >= 6)
        {
            escreva("O aluno teve um desempenho razoável na prova")
        }
        senao
        {
            escreva("O aluno teve um desempenho mau na prova")
        }
    }
}

```

## 2.5. Escolha-caso

Qual a melhor forma para programar um menu de, por exemplo, uma calculadora?

Esta tarefa poderia ser executada através de desvios condicionais **se** e **senao**, porém esta solução seria complexa e demorada. Assim, pode-se executar esta tarefa de uma maneira melhor, por meio de outro tipo de desvio condicional: uma estrutura de múltipla escolha, denominada **escolha** junto com o **caso**. Este comando é similar aos comandos **se** e **senao**, e reduz a complexidade do problema.

Apesar de suas similaridades com o **se**, ele possui algumas diferenças. Neste comando não é possível o uso de operadores lógicos, ele apenas trabalha com valores definidos, ou o valor é igual ou diferente. Além disso, o comando **escolha** e o **caso** têm alguns casos testes, e se a instrução **pare** não for colocada ao fim de cada um destes testes, o comando executará todos casos existentes.

A sintaxe da estrutura de múltipla escolha é composta do comando **escolha** a condição a ser testada e entre chaves se coloca os casos. A sintaxe para se criar um caso é a palavra reservada **caso**, o valor que a condição testada deve possuir, dois pontos e suas instruções. Lembre-se de terminá-las com o comando **pare**.

O comando **pare** evita que os blocos de comando seguinte sejam executados por engano. O **caso contrário** será executado caso nenhuma das expressões anteriores sejam executadas. O fluxograma abaixo representa um algoritmo que verifica se o valor da variável é igual a 0, 1 ou 2.

```

inteiro numero
escolha(numero)
{
    caso 1:
        //Instruções caso o numero for igual a 1
        pare

    caso 2:
        //Instruções caso o numero for igual a 2
        pare

    caso 50:
        //Instruções caso o numero for igual a 50
        pare

    caso contrario:
        //Instruções caso nenhum dos casos anteriores não seja verdadeiro
}

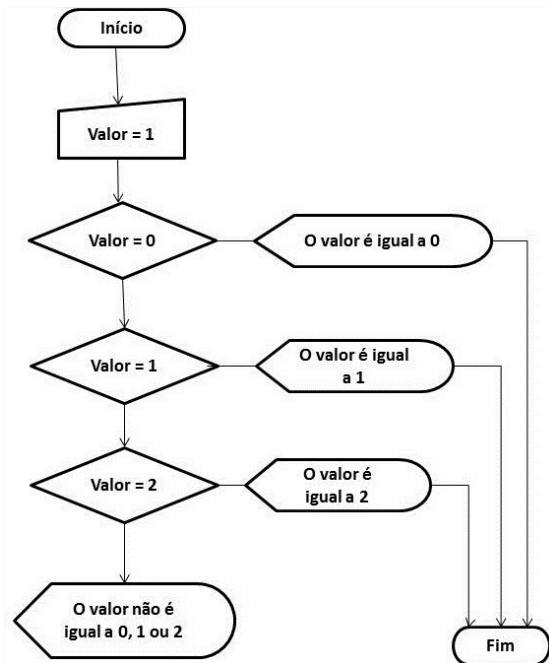
cadeia texto
escolha(texto)
{
    caso "sim":
        //Instruções caso o texto for igual a "sim"
        pare

    caso "nao":
        //Instruções caso o texto for igual a "nao"
}

```



O exemplo ao lado ilustra em português o mesmo algoritmo do fluxograma ao lado.



```

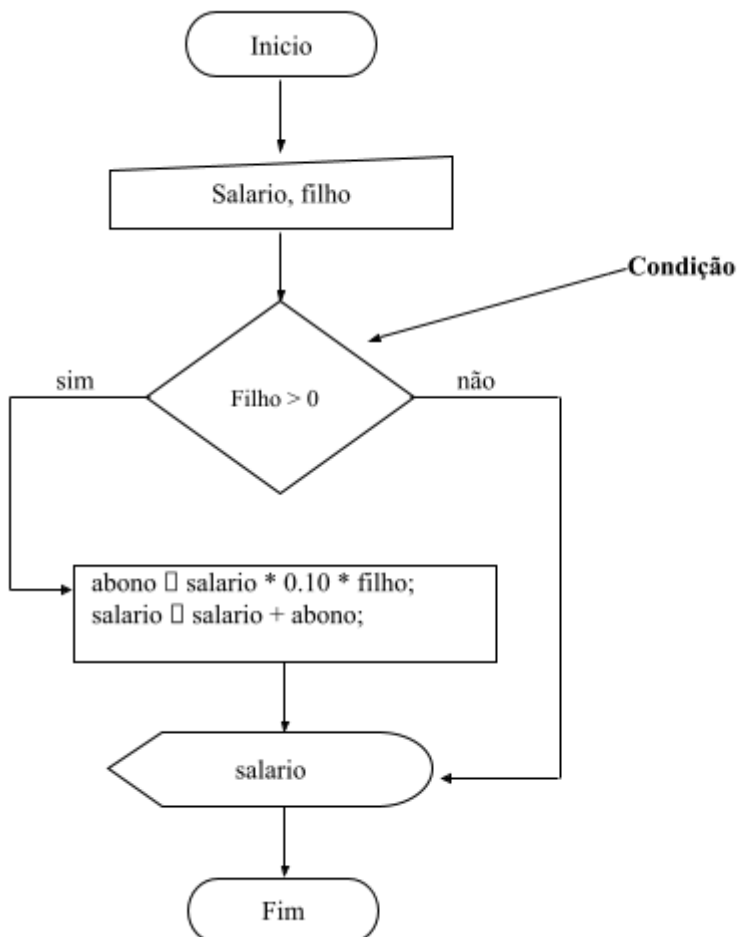
programa
{
  funcao inicio()
  {
    inteiro valor=1
    escolha (valor)
    {
      caso 0: //testa se o valor é igual a 0
      escreva ("o valor é igual a 0")
      pare

      caso 1: //testa se o valor é igual a 1
      escreva ("o valor é igual a 1")
      pare

      caso 2: //testa se o valor é igual a 2
      escreva ("o valor é igual a 2")
      pare

      caso contrario:
      escreva ("o valor não é igual a 0, 1 ou 2")
    }
  }
}
  
```

**Exemplo 1:** O algoritmo abaixo calcula o valor do abono família (10% do salário por filho) para o funcionário que tem filhos.



```

programa
{
    funcao inicio()
    {
        real salario, abono
        inteiro filhos
        escreva ("Informe o salário:")
        leia(salario)
        escreva ("Informe quantidade de filhos: ")
        leia (filhos)
        se ( filhos > 0 )
        {
            abono = salario * 0.10 * filhos
            salario = salario + abono
        }
        escreva ("O seu salário é: ", salario)
    }
}

```

**Exemplo 2:** O algoritmo abaixo verifica se o número informado pelo usuário é par ou ímpar.

```

programa
{
    funcao inicio()
    {
        inteiro numero
        escreva ("Informe um número inteiro: ")
        leia (numero)
        se(numero % 2 == 0)
        {
            escreva ("O número é par")
        }
        senao
        {
            escreva ("O número é ímpar")
        }
    }
}

```

**Exemplo 3** – Elabore um algoritmo que após ler 2 números informe qual o menor número ou se os números são iguais.

```

programa
{
    funcao inicio()
    {
        inteiro x, z
        escreva ("Digite o número inteiro (x): ")
        leia ( x )
        escreva ("Digite o número inteiro (y): ")
        leia ( y )
    }
}

```

```

    se ( x < y )
    {
        escreva ("Menor número = ", x )
    }
    senao se (x > y)
    {
        escreva ("Menor número = ", y )
    }
    senao
    {
        escreva ("Números iguais")
    }
}

```

## Resolva os algoritmos propostos

1. Elabore um algoritmo para solicitar ao usuário um número e informar se este é par ou impar.
2. Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um algoritmo que calcule seu peso ideal, utilizando as seguintes fórmulas:
  - para homens:  $(72,7 \cdot h) - 58$ ;
  - para mulheres  $(62,1 \cdot h) - 44,7$ .
3. Solicitar ao usuário 03 números. Informar o maior entre eles.
4. Dado as seguintes informações de um funcionário: Nome, idade cargo e o seu salário bruto considere:
  - a) O salário bruto teve um reajuste de 38%.
  - a) O funcionário receberá uma gratificação de 20% do salário bruto.
  - a) O Salário total é descontado em 15%

Faça um algoritmo para imprimir o Nome, idade, cargo, salário bruto e salário líquido.
5. Faça um algoritmo que leia os valores A, B, C e diga se a soma de  $A + B$  é menor que C.
6. Faça um algoritmo que leia dois valores inteiros A e B se os valores forem iguais deverá se somar os dois, caso contrário multiplique A por B ao final do cálculo atribuir o valor para uma variável C.
7. Construa um algoritmo que leia o nome, a idade, número de dependentes, o salário e o sexo de um funcionário de uma empresa. Analise cada questão abaixo. Seu algoritmo deverá dar uma solução a cada uma delas, calculando e escrevendo o salário obtido:
  - Quando o funcionário for homem e possuir número de dependentes acima de 7 seu salário deverá ter um aumento de 25%.
  - Quando o funcionário for um homem com idade superior a 55 anos seu salário deverá ter um aumento de 15%.

- Quando o funcionário for uma mulher, com idade superior a 40 anos ou número de dependente acima de 5, deverá ter um aumento de 30%.
  - Os demais funcionários deverão ter um aumento de 5%;
  - Se o usuário informar um sexo inválido seu algoritmo deverá detectar e informar ao usuário.
8. Construa um algoritmo que seja capaz de ler o preço de 3 produtos comprados. Calcular o preço total da compra. Sabendo-se que sobre o preço total da compra, o cliente terá um desconto de 15%. Seu algoritmo deverá calcular e escrever o preço final desta compra após ser dado o desconto.
  9. Construa um algoritmo que leia o nome, o sexo e altura de duas pessoas. Quando a pessoa for do sexo feminino calcule e escreva o dobro da altura desta pessoa. Caso ela seja no sexo masculino calcule e escreva a metade da altura desta pessoa.
  10. Construa um algoritmo que seja capaz de ler o preço de 3 produtos comprados. Além disso, o comprador deverá informar se deseja comprar à vista ou com um cheque para 30 dias. Peça ao usuário para digitar um (1) quando for comprar à vista e zero (0) quando for dar um cheque para 30 dias.. Calcular e escrever o preço total da compra. Seu algoritmo deverá calcular e escrever também o preço final desta compra, sabendo-se que para compras à vista tem-se um desconto de 10 % e para compras com 30 dias no cheque tem-se um aumento de 5%.
  11. Construa um algoritmo que leia o salário e o tempo de serviço de uma pessoa. Quando esta pessoa tiver um salário menor que 150,00 e tempo de casa inferior a 2 anos lhe será dado 7 dias de folga e um aumento de 10%. Calcule e escreva o novo salário e comunique ( escreva) a quantidade de dias de folga que terá direito. Quando a pessoa ganhar mais ou igual a 150,00 e tiver tempo de casa superior a 3 anos lhe será dado 12 dias de folga e um aumento de 8%. Calcule e escreva o novo salário e comunique ( escreva) a quantidade de dias de folga que terá direito. As demais pessoas só terão direito a 15 dias de folga, sem qualquer aumento.
  12. Escreva um programa que leia um número inteiro maior do que zero e devolva, na tela, a soma de todos os seus algarismos. Por exemplo, o número 251 corresponderá ao valor 8 ( $2 + 5 + 1$ ). Se o número lido não for maior do que zero, o programa terminará com a mensagem “Número inválido”.
  13. A nota final de um estudante é calculada a partir de três notas atribuídas entre o intervalo de 0 até 10, respectivamente, a um trabalho de laboratório, a uma avaliação semestral e a um exame final. A média das três notas mencionadas anteriormente obedece aos pesos: Trabalho de Laboratório: 2; Avaliação Semestral: 3; Exame Final: 5. De acordo com o resultado, mostre na tela se o aluno está reprovado (média entre 0 e 2,9), de recuperação (entre 3 e 4,9) ou se foi aprovado. Faça todas as verificações necessárias.
  14. Escreva um programa que leia um inteiro entre 1 e 7 e imprima o dia da semana correspondente a este número. Isto é, domingo se 1, segunda-feira se 2, e assim por diante.
  15. Dados três valores, A,B,C, verificar se eles podem ser valores dos lados de um triângulo e, se forem, se é um triângulo escaleno, equilátero ou isóscele, considerando os seguintes conceitos:
    - O comprimento de cada lado de um triângulo é menor do que a soma dos outros dois lados.
    - Chama-se equilátero o triângulo que tem três lados iguais.
    - Denominam-se isósceles o triângulo que tem o comprimento de dois lados iguais.
    - Recebe o nome de escaleno o triângulo que tem os três lados diferentes.

16. Escreva o menu de opções abaixo. Leia a opção do usuário e execute a operação escolhida. Escreva uma mensagem de erro se a opção for inválida.  
Escolha a opção:  
1- Soma de 2 números.  
2- Diferença entre 2 números (maior pelo menor).  
3- Produto entre 2 números.  
4- Divisão entre 2 números (o denominador não pode ser zero).
17. Uma empresa vende o mesmo produto para quatro diferentes estados. Cada estado possui uma taxa diferente de imposto sobre o produto (MG 7%; SP 12%; RJ 15%; MS 8%). Faça um programa em que o usuário entre com o valor e o estado destino do produto e o programa retorne o preço final do produto acrescido do imposto do estado em que ele será vendido. Se o estado digitado não for válido, mostrar uma mensagem de erro.
18. Faça um programa que receba a altura e o peso de uma pessoa. De acordo com a tabela a seguir, verifique e mostre qual a classificação dessa pessoa.

Altura	Peso		
	Até 60	Entre 60 e 90 (Inclusive)	Acima de 90
Menor que 1,20	A	D	G
De 1,20 a 1,70	B	E	H
Maior que 1,70	C	F	I

19. Escrever um programa que leia o código do produto escolhido do cardápio de uma lanchonete e a quantidade. O programa deve calcular o valor a ser pago por aquele lanche. Considere que a cada execução somente será calculado um pedido. O cardápio da lanchonete segue o padrão abaixo:

Especificação	Código	Preço
Cachorro Quente	100	1.20
Bauru Simples	101	1.30
Bauru com Ovo	102	1.50
Hamburguer	103	1.20
Cheeseburger	104	1.70
Suco	105	2.20
Refrigerante	106	1.00

20. Escrever um programa que informe a quantidade total de calorias de uma refeição a partir do usuário que deverá informar o prato, a sobremesa e a bebida (veja a tabela a seguir).

Prato	Calorias	Sobremesa	Calorias	Bebida	Calorias
Vegetariano	180 cal	Abacaxi	75 cal	Chá	20 cal
Peixe	230 cal	Sorvete <i>diet</i>	110 cal	Suco de laranja	70 cal
Frango	250 cal	Mouse <i>diet</i>	170 cal	Suco de melão	100 cal
Carne	350 cal	Mouse chocolate	200 cal	Refrigerante <i>diet</i>	65 cal

Sugestão: enumere cada opção de prato, sobremesa e bebida. Ou seja:

**Prato:** 1 - vegetariano, 2 – Peixe, 3 – Frango, 4 – Carne;

**Sobremesa:** 1 – Abacaxi, 2 – Sorvete *diet*, 3 – Mouse *diet*, 4 – Mouse chocolate;

**Bebida:** 1 – Chá, 2 - Suco de laranja, 3 – Suco de melão, 4 – Refrigerante *diet*.

21. Criar um algoritmo em PORTUGOL que leia o número correspondente ao mês atual e os dígitos (somente os quatro números) de uma placa de veículo, e através do número finalizador da placa (algarismo da casa das unidades) determine se o IPVA do veículo vence no mês corrente.

Final 1 – mês (1) – Janeiro	Final 6 – mês (6) – Junho
Final 2 – mês (2) – Fevereiro	Final 7 – mês (7) – Julho
Final 3 – mês (3) – Março	Final 8 – mês (8) – Agosto
Final 4 – mês (4) – Abril	Final 9 – mês (9) – Setembro
Final 5 – mês (5) – Maio	Final 0 – mês (10) – Outubro

22. Um banco concederá um crédito especial aos seus clientes, variável com o saldo médio no último ano. Faça um algoritmo que leia o saldo médio de um cliente e calcule o valor do crédito de acordo com a tabela abaixo. Mostre uma mensagem informando o saldo médio e o valor do crédito. (use o comando caso-de e não faça repetições)

<b>Saldo médio .....</b>	<b>Percentual</b>
de 0 a 200.....	nenhum crédito
de 201 a 400.....	20% do valor do saldo médio
de 401 a 600.....	30% do valor do saldo médio
acima de 601 .....	40% do valor do saldo médio

23. Um vendedor precisa de um algoritmo que calcule o preço total devido por um cliente. O algoritmo deve receber o código de um produto e a quantidade comprada e calcular o preço total, usando a tabela abaixo. Mostre uma mensagem no caso de código inválido.

<b>Código</b>	<b>Preço Unitário</b>
'ABCD'	R\$ 5,30
'XYPK'	R\$ 6,00
'KLMP'	R\$ 3,20
'QRST'	R\$ 2,50

24. Escrever um algoritmo que lê a hora de início e hora de término de um jogo, ambas subdivididas em dois valores distintos: horas e minutos. Calcular e escrever a duração do jogo, também em horas e minutos, considerando que o tempo máximo de duração de um jogo é de 24 horas e que o jogo pode iniciar em um dia e terminar no dia seguinte.

25. Uma empresa paga a seus funcionários R\$ 1,00 de comissão para cada produto vendido, entretanto, se forem vendidos mais de 250 produtos, o valor aumenta para R\$ 1,50. Se a quantidade for superior a 500 produtos, o valor da comissão sobe para R\$ 2,00. Monte um programa para ler o nome de um funcionário e a quantidade de produtos que ele vendeu. Exiba o nome do funcionário e o total de comissão que ele vai receber.