

4.1 Estrutura condicional em algoritmos

A estrutura condicional em algoritmos pode ser simples ou composta.

4.1.1 Estrutura condicional simples

```
SE condição
ENTÃO comando
```

O comando só será executado se a condição for verdadeira. Uma condição é uma comparação que possui dois valores possíveis: verdadeiro ou falso.

```
SE condição
ENTÃO INÍCIO
    comando1
    comando2
    comando3
FIM
```

Os comandos 1, 2 e 3 só serão executados se a condição for verdadeira. As palavras INÍCIO e FIM serão necessárias apenas quando dois ou mais comandos forem executados.

4.1.2 Estrutura condicional composta

```
SE condição
ENTÃO comando1
SENÃO comando2
```

Se a condição for verdadeira, será executado o comando1; caso contrário, será executado o comando2.

```
SE condição
ENTÃO INÍCIO
    comando1
    comando2
FIM

SENÃO INÍCIO
    comando3
    comando4
FIM
```

Se a condição for verdadeira, o comando1 e o comando2 serão executados; caso contrário, o comando3 e o comando4 serão executados.

4.2 Estrutura condicional em PASCAL

A seguir, serão apresentadas três estruturas condicionais em PASCAL (simples, composta e case) e os operadores lógicos.

4.2.1 Estrutura condicional simples

```
IF condição  
THEN comando;
```

O comando só será executado se a condição for verdadeira. Uma condição é uma comparação que possui dois valores possíveis: verdadeiro ou falso.

```
IF condição  
THEN BEGIN  
    comando1;  
    comando2;  
    comando3;  
END;
```

Os comandos 1, 2 e 3 só serão executados se a condição for verdadeira.

4.2.2 Estrutura condicional composta

```
IF condição  
THEN comando1  
ELSE comando2;
```

Se a condição for verdadeira, será executado o comando1; caso contrário, será executado o comando2.

```
IF condição  
THEN BEGIN  
    comando1;  
    comando2;  
END  
ELSE BEGIN  
    comando3;  
    comando4;  
END;
```

Se a condição for verdadeira, o comando1 e o comando2 serão executados; se for falsa, o comando3 e o comando4 serão executados.



Observação

Antes do comando ELSE não existe ponto e vírgula.

4.2.3 Estrutura case

Em alguns programas, existem situações mutuamente exclusivas, isto é, se uma situação for executada, as demais não serão. Quando este for o caso, um comando seletivo será o mais indicado, e esse comando, em PASCAL, tem a seguinte sintaxe:

```

CASE seletor OF
  lista de alvos1: comando1;
  lista de alvos2: comando2;
  alvo3: comando3;
  alvo4: BEGIN
    comando4;
    comando5;
  END;
END;

```

Se o seletor atingir a lista de alvos1, o comando1 será executado; se atingir a lista de alvos2, o comando2 será executado; se atingir o alvo3, o comando3 será executado; e, se atingir o alvo4, então, o comando4 e o comando5 serão executados. Se nenhum alvo for atingido, nada será executado.

```

CASE seletor OF
  lista de alvos1: BEGIN
    comando1;
    comando2;
  END;
  lista de alvos2: comando3;
  ELSE comando4;
END;

```

Na estrutura CASE acima, se nenhum alvo for atingido os comandos da estrutura ELSE serão executados.

Exemplo:

```

program teste;
uses crt;
var i: integer;
begin
  clrscr;
  writeln('Digite um número');
  readln(i);
  case i of
    1: writeln('Número 1');
    2,5,6: writeln('Número 2 ou número 5 ou número 6');
    7..10: writeln('Número entre 7 e 10');
    else writeln('outro número');
  end;
  readln;
end.

```

A restrição da estrutura case é que o seletor só pode ser uma variável do tipo char, integer ou boolean.

4.2.4 Operadores lógicos

Os principais operadores lógicos são: AND, OR e NOT, que significam *e*, *ou*, *não* e são usados para conjunção, disjunção e negação, respectivamente.

TABELA E	TABELA OU	TABELA NÃO
V e V = V	V ou V = V	Não V = F
V e F = F	V ou F = V	Não F = V
F e V = F	F ou V = V	
F e F = F	F ou F = F	

Observações

Na linguagem PASCAL, quando existe mais de uma condição, elas devem estar entre parênteses.

Exemplos:

```
IF x = 3
THEN WRITELN('Número igual a 3');
```

No exemplo, existe apenas uma condição, logo, os parênteses são opcionais.

```
IF (X > 5) AND (X < 10)
THEN WRITELN('Número entre 5 e 10');
```

No exemplo anterior, existe mais de uma condição, logo, os parênteses são obrigatórios, ou seja, cada condição deve estar entre parênteses.

```
IF ((X = 5) AND (Y = 2)) OR (Y = 3)
THEN WRITELN('X é igual a 5 e Y é igual a 2, ou Y é igual a 3');
```

No exemplo anterior, existe mais de uma condição e mais de um tipo de operador lógico, logo, além dos parênteses de cada condição, devem existir ainda parênteses que indiquem a prioridade de execução das condições. Nesse exemplo, as condições com o operador AND, ou seja, ((X = 5) AND (Y = 2)), serão testadas e seu resultado será testado com a condição OR (Y = 3).

```
IF (X = 5) AND ((Y = 2) OR (Y = 3))
THEN WRITELN('X é igual a 5, e Y é igual a 2 ou Y é igual a 3');
```

Neste exemplo, existe mais de uma condição e mais de um tipo de operador lógico, logo, além dos parênteses de cada condição, devem existir ainda parênteses que indiquem a prioridade de execução das condições. Aqui, as condições com o operador OR, ou seja, ((Y = 2) OR (Y = 3)), serão testadas, e seu resultado será testado com a condição AND (X = 5).

4.3 Estrutura condicional em C/C++

A seguir, serão apresentadas três estruturas condicionais em C/C++ (simples, composta e case) e os operadores lógicos.

4.3.1 Estrutura condicional simples

```
if (condição)
    comando;
```

O comando só será executado se a condição for verdadeira. Uma condição é uma comparação que possui dois valores possíveis: verdadeiro ou falso.

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

Em C/C++, torna-se obrigatória a utilização de chaves quando existe mais de um comando a executar. Os comandos entre chaves { } só serão executados se a condição for verdadeira.

4.3.2 Estrutura condicional composta

```
if (condição)
    comando1;
else
    comando2;
```

Se a condição for verdadeira, será executado o comando1; se for falsa, será executado o comando2.

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

Se a condição for verdadeira, o comando1 e o comando2 serão executados; caso contrário, o comando3 e o comando4 serão executados.

4.3.3 Estrutura case

Em alguns programas, existem situações mutuamente exclusivas, isto é, se uma situação for executada, as demais não serão. Quando este for o caso, um comando seletivo é o mais indicado. Esse comando em C/C++ tem a seguinte sintaxe:

```
switch (variável)
{
    case valor1: lista de comandos;
        break;
    case valor2: lista de comandos;
        break;
        ....
    default: lista de comandos;
}
```

O comando `switch(variável)` avalia o valor de uma variável para decidir qual case será executado. Cada case está associado a UM possível valor da variável, que deve ser, obrigatoriamente, do tipo `char`, `unsigned char`, `int`, `unsigned int`, `short int`, `long` ou `unsigned long`.

O comando `break` deve ser utilizado para impedir a execução dos comandos definidos nos cases subsequentes. Quando o valor da variável não coincidir com aqueles especificados nos cases, será executado então o `default`.

Exemplo:

```
#include <stdio.h>
int main()
{
    int i;
    printf("Digite um número ");
    scanf("%d%c",&i);
    switch (i)
    {
        case 1:
            printf("Número 1");
            break;
        case 2:
```

```

    printf("Número 2");
    break;
default:
    printf("Número diferente de 1 e de 2");
}
getchar();
return 0;
}

```

4.3.4 Operadores lógicos

Os principais operadores lógicos são: `&&`, `||` e `!`, que significam *e*, *ou*, *não* e são usados para conjunção, disjunção e negação, respectivamente.

TABELA E	TABELA OU	TABELA NÃO
V e V = V	V ou V = V	Não V = F
V e F = F	V ou F = V	Não F = V
F e V = F	F ou V = V	
F e F = F	F ou F = F	



Observações

Na linguagem C/C++, todas as condições devem estar entre parênteses.

Exemplos:

```

if (x == 3)
printf("Número igual a 3");

```

No exemplo anterior, existe apenas uma condição que, obrigatoriamente, deve estar entre parênteses.

```

if (X > 5 && X < 10)
printf("Número entre 5 e 10");

```

No exemplo anterior, existe mais de uma condição, as quais, obrigatoriamente, devem estar entre parênteses.

```

if ((X == 5 && Y == 2) || Y == 3)
printf("X é igual a 5 e Y é igual a 2, ou Y é igual a 3");

```

No exemplo anterior, existe mais de uma condição e mais de um tipo de operador lógico, logo, além dos parênteses que envolvem todas as condições, devem existir ainda parênteses que indiquem a prioridade de execução das condições. Aqui, as condições com o operador `&&`, ou seja, `(X == 5 && Y == 2)`, serão testadas. Seu resultado será testado com a condição `|| Y == 3`.

```

if (X == 5 && (Y == 2 || Y == 3))
printf("X é igual a 5, e Y é igual a 2 ou Y é igual a 3");

```

No exemplo anterior, existe mais de uma condição e mais de um tipo de operador lógico, portanto, além dos parênteses que envolvem todas as condições, devem existir ainda parênteses que indiquem a prioridade de execução das condições. Nesse exemplo, as condições com o operador `||`, ou seja, `(Y == 2 || Y == 3)`, serão testadas. Seu resultado será testado com a condição `&& X == 5`.

4.4 Estrutura condicional em JAVA

A seguir, serão apresentadas três estruturas condicionais em JAVA (simples, composta e case) e os operadores lógicos.

4.4.1 Estrutura condicional simples

```
if (condição)
    comando;
```

O comando só será executado se a condição for verdadeira. Uma condição é uma comparação que possui dois valores possíveis: verdadeiro ou falso.

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

Em JAVA, torna-se obrigatória a utilização de chaves quando existe mais de um comando a executar. Os comandos entre chaves { } só serão executados se a condição for verdadeira.

4.4.2 Estrutura condicional composta

```
if (condição)
    comando1;
else
    comando2;
```

Se a condição for verdadeira, será executado o comando1; caso contrário, será executado o comando2.

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

Se a condição for verdadeira, o comando1 e o comando2 serão executados; se for falsa, o comando3 e o comando4 serão executados.

4.4.3 Estrutura case

Em alguns programas, existem situações mutuamente exclusivas, isto é, se uma situação for executada, as demais não serão. Quando este for o caso, um comando seletivo é o mais indicado, e esse comando, em JAVA, tem a seguinte sintaxe:

```
switch (variável)
{
    case valor1: lista de comandos;
        break;
    case valor2: lista de comandos;
        break;
        ....
    default: lista de comandos;
}
```

O comando switch (variável) analisa o valor de uma variável para decidir qual case será executado. Cada case está associado a UM possível valor da variável, que deve ser obrigatoriamente do tipo int, short, byte ou char.

O comando `break` deve ser utilizado para impedir a execução dos comandos definidos nos `cases` subsequentes.

Quando o valor da variável não coincidir com aqueles especificados nos `cases`, será executado, então, o `default`.

Exemplo:

```
import java.io.*;
import java.util.*;
class teste
{
    public static void main(String args[])
    {
        int x;
        Scanner dado;

        System.out.println("Digite um número ");
        dado = new Scanner(System.in);
        x = dado.nextInt();
        switch (x)
        {
            case 1: System.out.println("Número 1");
                    break;
            case 2: System.out.println("Número 2");
                    break;
            default: System.out.println("Outro número");
        }
    }
}
```

4.4.4 Operadores lógicos

Os principais operadores lógicos são: `&&`, `||` e `!`, que significam *e*, *ou*, *não* e são usados para conjunção, disjunção e negação, respectivamente.

TABELA E	TABELA OU	TABELA NÃO
V e V = V	V ou V = V	Não V = F
V e F = F	V ou F = V	Não F = V
F e V = F	F ou V = V	
F e F = F	F ou F = F	



Observações

Na linguagem JAVA, todas as condições devem estar entre parênteses.

Exemplos:

```
if (x == 3)
    System.out.println("Número igual a 3");
```

No exemplo anterior, existe apenas uma condição que, obrigatoriamente, deve estar entre parênteses.

```
if (X > 5 && X < 10)
    System.out.println("Número entre 5 e 10");
```


No exemplo anterior, existe mais de uma condição, as quais, obrigatoriamente, devem estar entre parênteses.

```
if ((X == 5 && Y == 2) || Y == 3)
    System.out.println("X é igual a 5 e Y é igual a 2, ou Y é igual a 3");
```

No exemplo anterior, existe mais de uma condição e mais de um tipo de operador lógico, portanto, além dos parênteses que envolvem todas as condições, devem existir ainda parênteses que indiquem a prioridade de execução das condições. Nesse exemplo, as condições com o operador &&, ou seja, (X == 5 && Y == 2), serão testadas, e seu resultado será testado com a condição || Y == 3.

```
if (X == 5 && (Y == 2 || Y == 3))
    System.out.println("X é igual a 5, e Y é igual a 2 ou Y é igual a 3");
```

No exemplo anterior, existe mais de uma condição e mais de um tipo de operador lógico, portanto, além dos parênteses que envolvem todas as condições, devem existir ainda parênteses que indiquem a prioridade de execução das condições. Nesse exemplo, as condições com o operador ||, ou seja, (Y == 2 || Y == 3), serão testadas, e seu resultado será testado com a condição && X == 5.

EXERCÍCIOS RESOLVIDOS

1. A nota final de um estudante é calculada a partir de três notas atribuídas, respectivamente, a um trabalho de laboratório, a uma avaliação semestral e a um exame final. A média das três notas mencionadas obedece aos pesos a seguir:

NOTA	PESO
Trabalho de laboratório	2
Avaliação semestral	3
Exame final	5

Faça um programa que receba as três notas, calcule e mostre a média ponderada e o conceito que segue a tabela:

MÉDIA PONDERADA			CONCEITO
8,0	● — ○	10,0	A
7,0	● — ○	8,0	B
6,0	● — ○	7,0	C
5,0	● — ○	6,0	D
0,0	● — ○	5,0	E

ALGORITMO SOLUÇÃO:

```
ALGORITMO
DECLARE nota_trab, aval_sem, exame, media NUMÉRICO
ESCREVA "Digite a nota do trabalho de laboratório: "
LEIA nota_trab
ESCREVA "Digite a nota da avaliação semestral: "
LEIA aval_sem
ESCREVA "Digite a nota do exame final: "
LEIA exame
media ← (nota_trab * 2 + aval_sem * 3 + exame * 5) / 10
ESCREVA "Média ponderada: " , media
SE media >= 8 E media <= 10
```

```

        ENTÃO ESCRIVA "Obteve conceito A"
SE media >= 7 E media < 8
        ENTÃO ESCRIVA "Obteve conceito B"
SE media >= 6 E media < 7
        ENTÃO ESCRIVA "Obteve conceito C"
SE media >= 5 E media < 6
        ENTÃO ESCRIVA "Obteve conceito D"
SE media >= 0 E media < 5
        ENTÃO ESCRIVA "Obteve conceito E"
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX1_A.PAS e \EXERC\CAP4\PASCAL\EX1_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX1_B.PAS e \EXERC\CAP4\PASCAL\EX1_B.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX1_A.CPP e \EXERC\CAP4\C++\EX1_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX1_B.CPP e \EXERC\CAP4\C++\EX1_B.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX1_A.java e \EXERC\CAP4\JAVA\EX1_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX1_B.java e \EXERC\CAP4\JAVA\EX1_B.class

- 2.** Faça um programa que receba três notas de um aluno, calcule e mostre a média aritmética e a mensagem constante na tabela a seguir. Aos alunos que ficaram para exame, calcule e mostre a nota que deverão tirar para serem aprovados, considerando que a média exigida é 6,0.

MÉDIA ARITMÉTICA		MENSAGEM	
0,0	●————○	3,0	Reprovado
3,0	●————○	7,0	Exame
7,0	●————●	10,0	Aprovado

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE nota1, nota2, nota3, media, nota_exame NUMÉRICO
ESCREVA "Digite a primeira nota: "
LEIA nota1
ESCREVA "Digite a segunda nota: "
LEIA nota2
ESCREVA "Digite a terceira nota: "
LEIA nota3
media ← (nota1 + nota2 + nota3) / 3
ESCREVA "Média aritmética: ",media
SE media >= 0 E media < 3
    ENTÃO ESCRIVA "Reprovado"
SE media >= 3 E media < 7
    ENTÃO INÍCIO
        ESCRIVA "Exame"
        nota_exame ← 12 - media
    FIM

```

```

        ESCRIBA "Deve tirar nota", nota_exame, "para ser aprovado"
    FIM
SE media >= 7 E media <= 10
    ENTÃO ESCRIBA "Aprovado"
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX2_A.PAS e \EXERC\CAP4\PASCAL\EX2_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX2_B.PAS e \EXERC\CAP4\PASCAL\EX2_B.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX2_A.CPP e \EXERC\CAP4\C++\EX2_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX2_B.CPP e \EXERC\CAP4\C++\EX2_B.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX2_A.java e \EXERC\CAP4\JAVA\EX2_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX2_B.java e \EXERC\CAP4\JAVA\EX2_B.class

3. Faça um programa que receba dois números e mostre o maior.

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE num1, num2 NUMÉRICO
ESCREVA "Digite o primeiro número: "
LEIA num1
ESCREVA "Digite o segundo número: "
LEIA num2
SE num1 > num2
    ENTÃO ESCRIBA "O maior número é: ", num1
SE num2 > num1
    ENTÃO ESCRIBA "O maior número é: ", num2
SE num1 = num2
    ENTÃO ESCRIBA "Os números são iguais "
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX3_A.PAS e \EXERC\CAP4\PASCAL\EX3_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX3_B.PAS e \EXERC\CAP4\PASCAL\EX3_B.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX3_A.CPP e \EXERC\CAP4\C++\EX3_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX3_B.CPP e \EXERC\CAP4\C++\EX3_B.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX3_A.java e \EXERC\CAP4\JAVA\EX3_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX3_B.java e \EXERC\CAP4\JAVA\EX3_B.class

- 4.** Faça um programa que receba três números e mostre-os em ordem crescente. Suponha que o usuário digitará três números diferentes.

ALGORITMO SOLUÇÃO:

```
ALGORITMO
DECLARE num1, num2, num3 NUMÉRICO
ESCREVA "Digite o primeiro número: "
LEIA num1
ESCREVA "Digite o segundo número: "
LEIA num2
ESCREVA "Digite o terceiro número: "
LEIA num3
SE num1 < num2 E num1 < num3
    ENTÃO SE num2 < num3
        ENTÃO ESCREVA "A ordem crescente é: ", num1, "-", num2, "-", num3
        SENÃO ESCREVA "A ordem crescente é: ", num1, "-", num3, "-", num2
SE num2 < num1 E num2 < num3
    ENTÃO SE num1 < num3
        ENTÃO ESCREVA "A ordem crescente é: ", num2, "-", num1, "-", num3
        SENÃO ESCREVA "A ordem crescente é: ", num2, "-", num3, "-", num1
SE num3 < num1 E num3 < num2
    ENTÃO SE num1 < num2
        ENTÃO ESCREVA "A ordem crescente é: ", num3, "-", num1, "-", num2
        SENÃO ESCREVA "A ordem crescente é: ", num3, "-", num2, "-", num1
FIM_ALGORITMO.
```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX4_A.PAS e \EXERC\CAP4\PASCAL\EX4_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX4_B.PAS e \EXERC\CAP4\PASCAL\EX4_B.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX4_A.CPP e \EXERC\CAP4\C++\EX4_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX4_B.CPP e \EXERC\CAP4\C++\EX4_B.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX4_A.java e \EXERC\CAP4\JAVA\EX4_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX4_B.java e \EXERC\CAP4\JAVA\EX4_B.class

- 5.** Faça um programa que receba três números obrigatoriamente em ordem crescente e um quarto número que não siga essa regra. Mostre, em seguida, os quatro números em ordem decrescente. Suponha que o usuário digitará quatro números diferentes.

ALGORITMO SOLUÇÃO:

```
ALGORITMO
DECLARE num1, num2, num3, num4 NUMÉRICO
ESCREVA "Digite três números em ordem crescente: "
LEIA num1
LEIA num2
LEIA num3
ESCREVA "Digite um número (fora de ordem): "
LEIA num4
SE num4 > num3
```

```

    ENTÃO ESCREVA "A ordem decrescente é: ", num4, "-", num3, "-", num2, "-", num1
SE num4 > num2 E num4 < num3
    ENTÃO ESCREVA "A ordem decrescente é: ", num3, "-", num4, "-", num2, "-", num1
SE num4 > num1 E num4 < num2
    ENTÃO ESCREVA "A ordem decrescente é: ", num3, "-", num2, "-", num4, "-", num1
SE num4 < num1
    ENTÃO ESCREVA "A ordem decrescente é: ", num3, "-", num2, "-", num1, "-", num4
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX5_A.PAS e \EXERC\CAP4\PASCAL\EX5_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX5_B.PAS e \EXERC\CAP4\PASCAL\EX5_B.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX5_A.CPP e \EXERC\CAP4\C++\EX5_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX5_B.CPP e \EXERC\CAP4\C++\EX5_B.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX5_A.java e \EXERC\CAP4\JAVA\EX5_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX5_B.java e \EXERC\CAP4\JAVA\EX5_B.class

6. Faça um programa que receba um número inteiro e verifique se é par ou ímpar.**ALGORITMO** SOLUÇÃO:

```

ALGORITMO
DECLARE num, r NUMÉRICO
ESCREVA "Digite um número: "
LEIA num
r ← RESTO(num/2)
SE r = 0
    ENTÃO ESCREVA "O número é par"
SENÃO ESCREVA "O número é ímpar"
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX6_A.PAS e \EXERC\CAP4\PASCAL\EX6_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX6_B.PAS e \EXERC\CAP4\PASCAL\EX6_B.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX6_A.CPP e \EXERC\CAP4\C++\EX6_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX6_B.CPP e \EXERC\CAP4\C++\EX6_B.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX6_A.java e \EXERC\CAP4\JAVA\EX6_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX6_B.java e \EXERC\CAP4\JAVA\EX6_B.class

- 7.** Faça um programa que receba quatro valores: I, A, B e C. Desses valores, I é inteiro e positivo, A, B e C são reais. Escreva os números A, B e C obedecendo à tabela a seguir. Suponha que o valor digitado para I seja sempre um valor válido, ou seja, 1, 2 ou 3, e que os números digitados sejam diferentes um do outro.

VALOR DE I	FORMA A ESCRIVER
1	A, B e C em ordem crescente.
2	A, B e C em ordem decrescente.
3	O maior fica entre os outros dois números.

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE A, B, C, I NUMÉRICO
ESCREVA "Digite um valor para A:"
LEIA A
ESCREVA "Digite um valor para B:"
LEIA B
ESCREVA "Digite um valor para C:"
LEIA C
ESCREVA "Digite um valor para I (1, 2 ou 3):"
LEIA I
SE I=1
  ENTÃO INÍCIO
    SE A<B E A<C
      ENTÃO SE B<C
        ENTÃO ESCRIVA "A ordem crescente dos números é: ",A," -",B,"-",C
      SENÃO ESCRIVA "A ordem crescente dos números é: ",A," -",C,"-",B
    SE B<A E B<C
      ENTÃO SE A<C
        ENTÃO ESCRIVA "A ordem crescente dos números é: ",B,"-",A,"-",C
      SENÃO ESCRIVA "A ordem crescente dos números é: ",B,"-",C,"-",A
    SE C<A E C<B
      ENTÃO SE A<B
        ENTÃO ESCRIVA "A ordem crescente dos números é: ",C,"-",A,"-",B
      SENÃO ESCRIVA "A ordem crescente dos números é: ",C,"-",B,"-",A
    FIM
  SE I=2
    ENTÃO INÍCIO
      SE A>B E A>C
        ENTÃO SE B>C
          ENTÃO ESCRIVA "A ordem decrescente dos números é: ",A," -",B,"-",C
        SENÃO ESCRIVA "A ordem decrescente dos números é: ",A," -",C,"-",B
      SE B>A E B>C
        ENTÃO SE A>C
          ENTÃO ESCRIVA "A ordem decrescente dos números é: ",B," -",A,"-",C
        SENÃO ESCRIVA "A ordem decrescente dos números é: ",B," -",C,"-",A
      SE C>A E C>B
        ENTÃO SE A>B
          ENTÃO ESCRIVA "A ordem decrescente dos números é: ",C," -",A,"-",B
        SENÃO ESCRIVA "A ordem decrescente dos números é: ",C," -",B,"-",A
      FIM
    SE I=3

```

```

ENTÃO INÍCIO
    SE A>B E A>C
        ENTÃO ESCREVA "A ordem desejada é: ",B,"-",A,"-",C
    SE B>A E B>C
        ENTÃO ESCREVA "A ordem desejada é: ",A,"-",B,"-",C
    SE C>A E C>B
        ENTÃO ESCREVA "A ordem desejada é: ",A,"-",C,"-",B
    FIM
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO:

```
\EXERC\CAP4\PASCAL\EX7_A.PAS e \EXERC\CAP4\PASCAL\EX7_A.EXE
```

2ª SOLUÇÃO:

```
\EXERC\CAP4\PASCAL\EX7_B.PAS e \EXERC\CAP4\PASCAL\EX7_B.EXE
```

3ª SOLUÇÃO:

```
\EXERC\CAP4\PASCAL\EX7_C.PAS e \EXERC\CAP4\PASCAL\EX7_C.EXE
```

C/C++

1ª SOLUÇÃO:

```
\EXERC\CAP4\C++\EX7_A.CPP e \EXERC\CAP4\C++\EX7_A.EXE
```

2ª SOLUÇÃO:

```
\EXERC\CAP4\C++\EX7_B.CPP e \EXERC\CAP4\C++\EX7_B.EXE
```

3ª SOLUÇÃO:

```
\EXERC\CAP4\C++\EX7_C.CPP e \EXERC\CAP4\C++\EX7_C.EXE
```

JAVA

1ª SOLUÇÃO:

```
\EXERC\CAP4\JAVA\EX7_A.java e \EXERC\CAP4\JAVA\EX7_A.class
```

2ª SOLUÇÃO:

```
\EXERC\CAP4\JAVA\EX7_B.java e \EXERC\CAP4\JAVA\EX7_B.class
```

3ª SOLUÇÃO:

```
\EXERC\CAP4\JAVA\EX7_C.java e \EXERC\CAP4\JAVA\EX7_C.class
```

- 8.** Faça um programa que mostre o menu de opções a seguir, receba a opção do usuário e os dados necessários para executar cada operação.

Menu de opções:

1. Somar dois números.
2. Raiz quadrada de um número.

Digite a opção desejada:

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE num1, num2, soma, raiz, op NUMÉRICO
ESCREVA " MENU"
ESCREVA "1- Somar dois números"
ESCREVA "2- Raiz quadrada de um número"
ESCREVA "Digite sua opção: "
LEIA op
SE op = 1
    ENTÃO INÍCIO
        ESCREVA "Digite um valor para o primeiro número:"
        LEIA num1

```

```

        ESCRIVA "Digite um valor para o segundo número:"
        LEIA num2
        soma ← num1 + num2
        ESCRIVA "A soma de ",num1," e ",num2," é ",soma
    FIM
SE op = 2
    ENTÃO INÍCIO
        ESCRIVA "Digite um valor: "
        LEIA num1
        raiz ←  $\sqrt[3]{\text{num1}}$ 
        ESCRIVA "A raiz quadrada de ",num1," é ",raiz
    FIM
SE op ≠ 1 E op ≠ 2
    ENTÃO ESCRIVA "Opção inválida!"
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX8_A.PAS e \EXERC\CAP4\PASCAL\EX8_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX8_B.PAS e \EXERC\CAP4\PASCAL\EX8_B.EXE

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\PASCAL\EX8_C.PAS e \EXERC\CAP4\PASCAL\EX8_C.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX8_A.CPP e \EXERC\CAP4\C++\EX8_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX8_B.CPP e \EXERC\CAP4\C++\EX8_B.EXE

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\C++\EX8_C.CPP e \EXERC\CAP4\C++\EX8_C.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX8_A.java e \EXERC\CAP4\JAVA\EX8_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX8_B.java e \EXERC\CAP4\JAVA\EX8_B.class

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\JAVA\EX8_C.java e \EXERC\CAP4\JAVA\EX8_C.class

- 9.** Faça um programa que mostre a data e a hora do sistema nos seguintes formatos: DD/MM/AAAA – mês por extenso e hora:minuto.

ALGORITMO Solução:

```

ALGORITMO
DECLARE t, d, dia, mes, ano, hora, min NUMÉRICO
d ← OBTENHA_DATA;
dia ← OBTENHA_DIA(d)
mes ← OBTENHA_MÊS(d)
ano ← OBTENHA_ANO(d)
ESCREVA "Data Atual: " , dia, "/", mes, "/", ano, " - "
SE mes = 1
    ENTÃO ESCRIVA "janeiro"
SE mes = 2
    ENTÃO ESCRIVA "fevereiro"

```



```

SE mes = 3
    ENTÃO ESCRIVA "março"
SE mes = 4
    ENTÃO ESCRIVA "abril"
SE mes = 5
    ENTÃO ESCRIVA "maio"
SE mes = 6
    ENTÃO ESCRIVA "junho"
SE mes = 7
    ENTÃO ESCRIVA "julho"
SE mes = 8
    ENTÃO ESCRIVA "agosto"
SE mes = 9
    ENTÃO ESCRIVA "setembro"
SE mes = 10
    ENTÃO ESCRIVA "outubro"
SE mes = 11
    ENTÃO ESCRIVA "novembro"
SE mes = 12
    ENTÃO ESCRIVA "dezembro"
t ← OBTENHA_HORÁRIO;
hora ← OBTENHA_HORA(t)
min ← OBTENHA_MINUTO(t)
ESCREVA "Hora Atual: "
ESCREVA hora, ":" , min
FIM_ALGORITMO.

```

PASCAL SOLUÇÃO:

\EXERC\CAP4\PASCAL\EX9.PAS e \EXERC\CAP4\PASCAL\EX9.EXE

Na solução com a linguagem PASCAL, foram utilizados os comandos `getdate` e `gettime`, para obter a data e a hora do sistema operacional, respectivamente. O comando `getdate` retorna os valores do ano, mês, dia do mês e dia da semana da data do sistema operacional; as variáveis que receberão esses valores devem ser do tipo `word`. O comando `gettime` retorna os valores da hora, minuto, segundo e centésimo de segundo da hora do sistema operacional e as variáveis que receberão esses valores devem ser do tipo `word`. Para a utilização dos comandos `getdate` e `gettime`, é necessário utilizar a biblioteca `DOS`, ou seja, `USES DOS`.



Observação

O dia da semana é um número em que domingo vale 0; segunda-feira, 1; terça-feira, 2; quarta-feira, 3; quinta-feira, 4; sexta-feira, 5; e sábado, 6.

Exemplo:

```

GETDATE(ano, mes, dia, dia_semana);
GETTIME(hora, min, seg, cen_seg);

```

C/C++ SOLUÇÃO:

\EXERC\CAP4\C++\EX9.CPP e \EXERC\CAP4\C++\EX9.EXE

Na solução com a linguagem C/C++, foram utilizados os comandos `t = time(NULL)` e `data_hora = localtime(&t)` para obter a data e a hora do sistema operacional. Para a utilização desses comandos, é necessário utilizar a biblioteca `time.h`, ou seja, `#include <time.h>` e a declaração do tipo `time_t`.

Observação

Exemplo:

```

time_t t;
struct tm *data_hora;
int dia, mes, ano, hora, min, seg;
//Pega a data e a hora atual do sistema
t = time(NULL);
data_hora = localtime(&t);
dia = data_hora -> tm_mday;
mes = data_hora -> tm_mon + 1;
ano = data_hora -> tm_year + 1900;
hora = data_hora -> tm_hour;
min = data_hora -> tm_min;

// a declaração acima define um conjunto com 9 variáveis do tipo int.
// as principais variáveis são:
// tm_sec - segundos da hora do sistema (0 a 59)
// tm_min - minutos da hora do sistema (0 a 59)
// tm_hour - horas da hora do sistema (0 a 23)
// tm_mday - dia da data do sistema (1 a 31)
// tm_mon - mês da data do sistema (0 a 11)
// tm_year - ano da data do sistema (desde 1900)
// tm_wday - dia da semana da data do sistema (0 a 6, sendo 0 o domingo)

```

JAVA

SOLUÇÃO:

\EXERC\CAP4\JAVA\EX9.java e \EXERC\CAP4\JAVA\EX9.class

Na solução com a linguagem JAVA, foram utilizadas as classes Calendar e Date para empregar a data e a hora do sistema operacional.

Observação

Exemplo:

```

int dia, mes, ano, hora, min;
Calendar cal = Calendar.getInstance();
Date d = new Date();
cal.setTime(d);
dia = cal.get(Calendar.DAY_OF_MONTH);
mes = cal.get(Calendar.MONTH) + 1;
ano = cal.get(Calendar.YEAR);
hora = cal.get(Calendar.HOUR);
min = cal.get(Calendar.MINUTE);

```

- 10.** Faça um programa que determine a data cronologicamente maior entre duas datas fornecidas pelo usuário. Cada data deve ser composta por três valores inteiros, em que o primeiro representa o dia, o segundo, o mês e o terceiro, o ano.

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE d1,m1,a1,d2,m2,a2 NUMÉRICO
ESCREVA "Digite a primeira data"
ESCREVA " dia (dd): "
LEIA d1
ESCREVA " mês (mm): "
LEIA m1
ESCREVA " ano (aaaa): "
LEIA a1
ESCREVA "Digite a segunda data"
ESCREVA " dia (dd): "
LEIA d2
ESCREVA " mês (mm): "
LEIA m2
ESCREVA " ano (aaaa): "
LEIA a2
SE a1>a2
ENTÃO ESCREVA "A maior data é: ",d1,"-",m1,"-",a1
SENÃO SE a2>a1
    ENTÃO ESCREVA "A maior data é: ",d2,"-",m2,"-",a2
    SENÃO SE m1>m2
        ENTÃO ESCREVA "A maior data é: ",d1,"-",m1,"-",a1
        SENÃO SE m2>m1
            ENTÃO ESCREVA "A maior data é: ",d2,"-",m2,"-",a2
            SENÃO SE d1>d2
                ENTÃO ESCREVA "A maior data é: ",d1,"-",m1,"-",a1
                SENÃO SE d2>d1
                    ENTÃO ESCREVA "A maior data é: ",d2,"-",m2,"-",a2
                    SENÃO ESCREVA "As datas são iguais !"
FIM_ALGORITMO.

```

PASCAL SOLUÇÃO:

```
\EXERC\CAP4\PASCAL\EX10.PAS e \EXERC\CAP4\PASCAL\EX10.EXE
```

C/C++ SOLUÇÃO:

```
\EXERC\CAP4\C++\EX10.CPP e \EXERC\CAP4\C++\EX10.EXE
```

JAVA SOLUÇÃO:

```
\EXERC\CAP4\JAVA\EX10.java e \EXERC\CAP4\JAVA\EX10.class
```

- 11.** Faça um programa que receba a hora do início de um jogo e a hora do término (cada hora é composta por duas variáveis inteiras: hora e minuto). Calcule e mostre a duração do jogo (horas e minutos), sabendo que o tempo máximo de duração do jogo é de 24 horas e que ele pode começar em um dia e terminar no dia seguinte.

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE hora_i, min_i, hora_f, min_f, hora_d, min_d NUMÉRICO
ESCREVA "Digite o horário inicial"
ESCREVA "hora: "
LEIA hora_i

```

```

ESCREVA "minuto: "
LEIA min_i
ESCREVA "Digite o horário final "
ESCREVA "hora: "
LEIA hora_f
ESCREVA "minuto: "
LEIA min_f
SE min_i > min_f
    ENTÃO INÍCIO
        min_f ← min_f + 60
        hora_f ← hora_f - 1
    FIM
SE hora_i > hora_f
    ENTÃO hora_f ← hora_f + 24
min_d ← min_f - min_i;
hora_d ← hora_f - hora_i;
ESCREVA "O jogo durou ",hora_d," hora(s) e ",min_d," minuto(s)"
FIM_ALGORITMO.

```

PASCALSOLUÇÃO:

\EXERC\CAP4\PASCAL\EX11.PAS e \EXERC\CAP4\PASCAL\EX11.EXE

C/C++SOLUÇÃO:

\EXERC\CAP4\C++\EX11.CPP e \EXERC\CAP4\C++\EX11.EXE

JAVASOLUÇÃO:

\EXERC\CAP4\JAVA\EX11.java e \EXERC\CAP4\JAVA\EX11.class

- 12.** Faça um programa que receba o código correspondente ao cargo de um funcionário e seu salário atual e mostre o cargo, o valor do aumento e seu novo salário. Os cargos estão na tabela a seguir.

CÓDIGO	CARGO	PERCENTUAL
1	Escriturário	50%
2	Secretário	35%
3	Caixa	20%
4	Gerente	10%
5	Diretor	Não tem aumento

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE salario, aumento, novo_sal, cargo NUMÉRICO
ESCREVA "Digite o cargo do funcionário (1,2,3,4 ou 5)"
LEIA cargo
ESCREVA "Digite o valor do salário: "
LEIA salario
SE cargo = 1
    ENTÃO INÍCIO
        ESCREVA "O cargo é Escriturário"
        aumento ← salario * 50 / 100
        ESCREVA "O valor do aumento é: ", aumento

```

```

    novo_sal ← salario + aumento
    ESCRIVA "O novo salário é: ", novo_sal
FIM
SENÃO SE cargo = 2
    ENTÃO INÍCIO
        ESCRIVA "O cargo é Secretário"
        aumento ← salario * 35 / 100
        ESCRIVA "O valor do aumento é: ", aumento
        novo_sal ← salario + aumento
        ESCRIVA "O novo salário é: ", novo_sal
    FIM
SENÃO SE cargo = 3
    ENTÃO INÍCIO
        ESCRIVA "O cargo é Caixa"
        aumento ← salario * 20 / 100
        ESCRIVA "O valor do aumento é: ", aumento
        novo_sal ← salario + aumento
        ESCRIVA "O novo salário é: ", novo_sal
    FIM
SENÃO SE cargo = 4
    ENTÃO INÍCIO
        ESCRIVA "O cargo é Gerente"
        aumento ← salario * 10 / 100
        ESCRIVA "O valor do aumento é: ", aumento
        novo_sal ← salario + aumento
        ESCRIVA "O novo salário é: ", novo_sal
    FIM
SENÃO SE cargo = 5
    ENTÃO INÍCIO
        ESCRIVA "O cargo é Diretor"
        aumento ← salario * 0 / 100
        ESCRIVA "O valor do aumento é: ", aumento
        novo_sal ← salario + aumento
        ESCRIVA "O novo salário é: ", novo_sal
    FIM
SENÃO ESCRIVA "Cargo Inexistente !"
FIM_ALGORITMO.

```

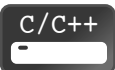


1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX12_A.PAS e \EXERC\CAP4\PASCAL\EX12_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\PASCAL\EX12_B.PAS e \EXERC\CAP4\PASCAL\EX12_B.EXE



1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX12_A.CPP e \EXERC\CAP4\C++\EX12_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\C++\EX12_B.CPP e \EXERC\CAP4\C++\EX12_B.EXE



1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX12_A.java e \EXERC\CAP4\JAVA\EX12_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\JAVA\EX12_B.java e \EXERC\CAP4\JAVA\EX12_B.class

13. Faça um programa que apresente o menu a seguir, permita ao usuário escolher a opção desejada, receba os dados necessários para executar a operação e mostre o resultado. Verifique a possibilidade de opção inválida e não se preocupe com restrições, como salário negativo.

Menu de opções:

1. Imposto
2. Novo salário
3. Classificação

Digite a opção desejada.

Na opção 1: receber o salário de um funcionário, calcular e mostrar o valor do imposto usando as regras a seguir.

SALÁRIO	PERCENTUAL DO IMPOSTO
Menor que R\$ 500,00	5%
De R\$ 500,00 (inclusive) a R\$ 850,00 (inclusive)	10%
Acima de R\$ 850,00	15%

Na opção 2: receber o salário de um funcionário, calcular e mostrar o valor do novo salário, usando as regras a seguir.

SALÁRIO	AUMENTO
Maior que R\$ 1.500,00	R\$ 25,00
De R\$ 750,00 (inclusive) a R\$ 1.500,00 (inclusive)	R\$ 50,00
De R\$ 450,00 (inclusive) a R\$ 750,00	R\$ 75,00
Menor que R\$ 450,00	R\$ 100,00

Na opção 3: receber o salário de um funcionário e mostrar sua classificação usando a tabela a seguir.

SALÁRIO	CLASSIFICAÇÃO
Até R\$ 700,00 (inclusive)	Mal remunerado
Maiores que R\$ 700,00	Bem remunerado

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE op, sal, imp, aum, novo_sal NUMÉRICO
ESCREVA "MENU DE OPÇÕES"
ESCREVA "1 – IMPOSTO"
ESCREVA "NOVO SALÁRIO"
ESCREVA "CLASSIFICAÇÃO"
ESCREVA "DIGITE A OPÇÃO DESEJADA"
LEIA op
SE op = 1
ENTÃO INÍCIO
    LEIA sal
    SE sal < 500
        ENTÃO imp ← sal * 5/100
    SE sal >= 500 E sal <= 850
        ENTÃO imp ← sal * 10/100
    SE sal > 850
        ENTÃO imp ← sal * 15/100

```

```

        ESCRIVA imp
    FIM
SE op = 2
ENTÃO INÍCIO
    LEIA sal
    SE sal > 1500
        ENTÃO aum ← 25
    SE sal >= 750 E sal <= 1500
        ENTÃO aum ← 50
    SE sal >= 450 E sal < 750
        ENTÃO aum ← 75
    SE sal < 450
        ENTÃO aum ← 100 novo_sal ← sal + aum
    ESCRIVA novo_sal
FIM
SE op = 3
ENTÃO INÍCIO
    LEIA sal
    SE sal <= 700
        ENTÃO ESCRIVA "Mal Remunerado"
    SE sal > 700
        ENTÃO ESCRIVA "Bem Remunerado"
FIM
SE op < 1 OU op > 3
ENTÃO ESCRIVA "Opção Inválida"
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

EXERC\CAP4\PASCAL\EX13_A.PAS e \EXERC\CAP4\PASCAL\EX13_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX13_B.PAS e \EXERC\CAP4\PASCAL\EX13_B.EXE

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\PASCAL\EX13_C.PAS e \EXERC\CAP4\PASCAL\EX13_C.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX13_A.CPP e \EXERC\CAP4\C++\EX13_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX13_B.CPP e \EXERC\CAP4\C++\EX13_B.EXE

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\C++\EX13_C.CPP e \EXERC\CAP4\C++\EX13_C.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

EXERC\CAP4\JAVA\EX13_A.java e \EXERC\CAP4\JAVA\EX13_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX13_B.java e \EXERC\CAP4\JAVA\EX13_B.class

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\JAVA\EX13_C.java e \EXERC\CAP4\JAVA\EX13_C.class

- 14.** Faça um programa que receba o salário inicial de um funcionário, calcule e mostre o novo salário, acrescido de bonificação e de auxílio escola.

SALÁRIO	BONIFICAÇÃO	SALÁRIO	AUXÍLIO ESCOLA
Até R\$ 500,00	5% do salário	Até R\$ 600,00	R\$ 150,00
Entre R\$ 500,00 e R\$ 1.200,00	12% do salário	Acima de R\$ 600,00	R\$ 100,00
Acima de R\$ 1.200,00	Sem bonificação		

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE sal, novo_sal, boni, aux NUMÉRICO
LEIA sal
SE sal <= 500
    ENTÃO boni ← sal * 5/100
    SENÃO SE sal <= 1200
        ENTÃO boni ← sal * 12/100
        SENÃO boni ← 0
SE sal <= 600
    ENTÃO aux ← 150
    SENÃO aux ← 100
novo_sal ← sal + boni + aux
ESCREVA novo_sal
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX14_A.PAS e \EXERC\CAP4\PASCAL\EX14_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX14_B.PAS e \EXERC\CAP4\PASCAL\EX14_B.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX14_A.CPP e \EXERC\CAP4\C++\EX14_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX14_B.CPP e \EXERC\CAP4\C++\EX14_B.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX14_A.java e \EXERC\CAP4\JAVA\EX14_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX14_B.java e \EXERC\CAP4\JAVA\EX14_B.class

- 15.** Faça um programa que receba o valor do salário mínimo, o número de horas trabalhadas, o número de dependentes do funcionário e a quantidade de horas extras trabalhadas. Calcule e mostre o salário a receber do funcionário de acordo com as regras a seguir:

- O valor da hora trabalhada é igual a 1/5 do salário mínimo.
- O salário do mês é igual ao número de horas trabalhadas multiplicado pelo valor da hora trabalhada.
- Para cada dependente, acrescentar R\$ 32,00.
- Para cada hora extra trabalhada, calcular o valor da hora trabalhada acrescida de 50%.
- O salário bruto é igual ao salário do mês mais o valor dos dependentes mais o valor das horas extras.
- Calcular o valor do imposto de renda retido na fonte de acordo com a tabela a seguir:

IRRF	SALÁRIO BRUTO
Isento	Inferior a R\$ 200,00
10%	De R\$ 200,00 até R\$ 500,00
20%	Superior a R\$ 500,00

- O salário líquido é igual ao salário bruto menos IRRF.
- A gratificação é de acordo com a tabela a seguir:

SALÁRIO LÍQUIDO	GRATIFICAÇÃO
Até R\$ 350,00	R\$ 100,00
Superior a R\$ 350,00	R\$ 50,00

- O salário a receber do funcionário é igual ao salário líquido mais a gratificação.

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE sal_min, nht, ndep, nhet NUMÉRICO
        sal_receber, vh, smes, vdep, vhe, imp NUMÉRICO
        sbruto, sliq, grat NUMÉRICO
LEIA sal_min, nht, ndep, nhet
vh ← 1/5 * sal_min
smes ← nht * vh
vdep ← 32 * ndep
vhe ← nhet * (vh + (vh * 50/100))
sbruto ← smes + vdep + vhe
SE sbruto < 200
    ENTÃO imp ← 0
SE sbruto >= 200 E sbruto <= 500
    ENTÃO imp ← sbruto * 10/100
SE sbruto > 500
    ENTÃO imp ← sbruto * 20/100
sliq ← sbruto - imp
SE sliq <= 350
    ENTÃO grat ← 100
SE sliq > 350
    ENTÃO grat ← 50
sal_receber ← sliq + grat
ESCREVA sal_receber
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX15_A.PAS e \EXERC\CAP4\PASCAL\EX15_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX15_B.PAS e \EXERC\CAP4\PASCAL\EX15_B.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX15_A.CPP e \EXERC\CAP4\C++\EX15_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX15_B.CPP e \EXERC\CAP4\C++\EX15_B.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX15_A.java e \EXERC\CAP4\JAVA\EX15_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX15_B.java e \EXERC\CAP4\JAVA\EX15_B.class

- 16.** Um supermercado deseja reajustar os preços de seus produtos usando o seguinte critério: o produto poderá ter seu preço aumentado ou diminuído. Para o preço ser alterado, o produto deve preencher pelo menos um dos requisitos a seguir:

VENDA MÉDIA MENSAL	PREÇO ATUAL	% DE AUMENTO	% DE DIMINUIÇÃO
< 500	< R\$ 30,00	10	–
≥ 500 e < 1.200	≥ R\$ 30,00 e < R\$ 80,00	15	–
≥ 1.200	≥ R\$ 80,00	–	20

Faça um programa que receba o preço atual e a venda média mensal do produto, calcule e mostre o novo preço.

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE pre, venda, novo_pre NUMÉRICO
LEIA pre, venda
SE venda<500 OU pre<30
ENTÃO novo_pre ← pre + 10/100 * pre
SENÃO SE (venda≥500 E venda<1200) OU (pre≥30 E pre<80)
    ENTÃO novo_pre ← pre + 15/100 * pre
    SENÃO SE venda≥1200 OU pre≥80
        ENTÃO novo_pre ← pre – 20/100 * pre
ESCREVA novo_pre
FIM_ALGORITMO.

```

PASCAL

SOLUÇÃO:

\EXERC\CAP4\PASCAL\EX16.PAS e \EXERC\CAP4\PASCAL\EX16.EXE

C/C++

SOLUÇÃO:

\EXERC\CAP4\C++\EX16.CPP e \EXERC\CAP4\C++\EX16.EXE

JAVA

SOLUÇÃO:

\EXERC\CAP4\JAVA\EX16.java e \EXERC\CAP4\JAVA\EX16.class

- 17.** Faça um programa para resolver equações do 2º grau.

```

ax² + bx + c = 0
A variável a deve ser diferente de zero.
Δ = b² – 4 * a * c
Δ < 0 → não existe raiz real
Δ = 0 → existe uma raiz real
x = (–b) / (2 * a)
Δ > 0 → existem duas raízes reais
x1 = (–b + √Δ) / (2 * a)
x2 = (–b – √Δ) / (2 * a)

```

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE a, b, c, delta, x1, x2 NUMÉRICO
LEIA a, b, c
SE a = 0
ENTÃO ESCREVA "Estes valores não formam uma equação de segundo grau"
SENÃO INÍCIO
    delta ← (b * b) - (4 * a * c)
    SE delta < 0
        ENTÃO ESCREVA "Não existe raiz real"
    SE delta = 0
        ENTÃO INÍCIO
            ESCREVA "Existe uma raiz real"
            x1 ← (- b) / (2 * a)
            ESCREVA x1
        FIM
    SE delta > 0
        ENTÃO INÍCIO
            ESCREVA "Existem duas raízes reais"
            x1 ← (- b +  $\sqrt{\text{delta}}$ ) / (2 * a)
            x2 ← (- b -  $\sqrt{\text{delta}}$ ) / (2 * a)
            ESCREVA x1, x2
        FIM
    FIM
FIM_ALGORITMO.

```

PASCAL SOLUÇÃO:

\EXERC\CAP4\PASCAL\EX17.PAS e \EXERC\CAP4\PASCAL\EX17.EXE

C/C++ SOLUÇÃO:

\EXERC\CAP4\C++\EX17.CPP e \EXERC\CAP4\C++\EX17.EXE

JAVA SOLUÇÃO:

\EXERC\CAP4\JAVA\EX17.java e \EXERC\CAP4\JAVA\EX17.class

18. Dados três valores X, Y e Z, verifique se eles podem ser os comprimentos dos lados de um triângulo e, se forem, verifique se é um triângulo equilátero, isósceles ou escaleno. Se eles não formarem um triângulo, escreva uma mensagem. Considere que:

- o comprimento de cada lado de um triângulo é menor que a soma dos outros dois lados;
- chama-se equilátero o triângulo que tem três lados iguais;
- denomina-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.

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE x, y, z NUMÉRICO
LEIA x, y, z
SE x < y + z E y < x + z E z < x + y
ENTÃO INÍCIO
    SE x = y E y = z
        ENTÃO ESCREVA "Triângulo Equilátero"

```

```

        SENÃO SE x = y OU x = z OU y = z
            ENTÃO ESCRIVA "Triângulo Isósceles"
        SENÃO SE x ≠ y E x ≠ z E y ≠ z
            ENTÃO ESCRIVA "Triângulo Escaleno"

    FIM
    SENÃO ESCRIVA "Essas medidas não formam um triângulo"
    FIM_ALGORITMO.

```



SOLUÇÃO:

\EXERC\CAP4\PASCAL\EX18.PAS e \EXERC\CAP4\PASCAL\EX18.EXE



SOLUÇÃO:

\EXERC\CAP4\C++\EX18.CPP e \EXERC\CAP4\C++\EX18.EXE



SOLUÇÃO:

\EXERC\CAP4\JAVA\EX18.java e \EXERC\CAP4\JAVA\EX18.class

- 19.** Faça um programa que receba a altura e o peso de uma pessoa. De acordo com a tabela a seguir, verifique e mostre a classificação dessa pessoa.

ALTURA	PESO		
	ATÉ 60	ENTRE 60 E 90 (INCLUSIVE)	ACIMA DE 90
Menores que 1,20	A	D	G
De 1,20 a 1,70	B	E	H
Maiores que 1,70	C	F	I

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE altura, peso  NUMÉRICO
LEIA altura, peso
SE altura < 1.20
    ENTÃO INÍCIO
        SE peso <= 60
            ENTÃO ESCRIVA "A"
        SE peso > 60 E peso <= 90
            ENTÃO ESCRIVA "D"
        SE peso > 90
            ENTÃO ESCRIVA "G"
    FIM
SE altura >= 1.20 E altura <= 1.70
    ENTÃO INÍCIO
        SE peso <= 60
            ENTÃO ESCRIVA "B"
        SE peso > 60 E peso <= 90
            ENTÃO ESCRIVA "E"
        SE peso > 90
            ENTÃO ESCRIVA "H"
    FIM
SE altura > 1.70
    ENTÃO INÍCIO
        SE peso <= 60
            ENTÃO ESCRIVA "C"
        SE peso > 60 E peso <= 90

```

```

        ENTÃO ESCREVA "F"
    SE peso > 90
        ENTÃO ESCREVA "I"
    FIM
FIM_ALGORITMO.

```

PASCAL 1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:
\EXERC\CAP4\PASCAL\EX19_A.PAS e \EXERC\CAP4\PASCAL\EX19_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:
\EXERC\CAP4\PASCAL\EX19_B.PAS e \EXERC\CAP4\PASCAL\EX19_B.EXE

C/C++ 1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:
\EXERC\CAP4\C++\EX19_A.CPP e \EXERC\CAP4\C++\EX19_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:
\EXERC\CAP4\C++\EX19_B.CPP e \EXERC\CAP4\C++\EX19_B.EXE

JAVA 1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:
\EXERC\CAP4\JAVA\EX19_A.java e \EXERC\CAP4\JAVA\EX19_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:
\EXERC\CAP4\JAVA\EX19_B.java e \EXERC\CAP4\JAVA\EX19_B.class

20. Faça um programa que receba:

- O código de um produto comprado, supondo que a digitação do código do produto seja sempre válida, isto é, um número inteiro entre 1 e 10.
- O peso do produto em quilos.
- O código do país de origem, supondo que a digitação do código seja sempre válida, isto é, um número inteiro entre 1 e 3.

Tabelas:

CÓDIGO DO PAÍS DE ORIGEM	IMPOSTO	CÓDIGO DO PRODUTO	PREÇO POR GRAMA
1	0%	1 a 4	10
2	15%	5 a 7	25
3	25%	8 a 10	35

Calcule e mostre:

- o peso do produto convertido em gramas;
- o preço total do produto comprado;
- o valor do imposto, sabendo que ele é cobrado sobre o preço total do produto comprado e depende do país de origem;
- o valor total, preço total do produto mais imposto.

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE cod_prod, peso_quilos NUMÉRICO
        cod_pais, peso_gramas, pre_total NUMÉRICO
        imposto, valor_total, pre_grama NUMÉRICO
LEIA cod_prod, peso_quilos, cod_pais
peso_gramas ← peso_quilos * 1000
ESCREVA peso_gramas
SE cod_prod >= 1 E cod_prod <= 4

```

```

    ENTÃO pre_grama ← 10
SE cod_prod >= 5 E cod_prod <= 7
    ENTÃO pre_grama ← 25
SE cod_prod >= 8 E cod_prod <= 10
    ENTÃO pre_grama ← 35
pre_total ← peso_gramas * pre_grama
ESCREVA pre_total
SE cod_pais = 1
    ENTÃO imposto ← 0
SE cod_pais = 2
    ENTÃO imposto ← pre_total * 15/100
SE cod_pais = 3
    ENTÃO imposto ← pre_total * 25/100
ESCREVA imposto
valor_total ← pre_total + imposto
ESCREVA valor_total
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\PASCAL\EX20_A.PAS e \EXERC\CAP4\PASCAL\EX20_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX20_B.PAS e \EXERC\CAP4\PASCAL\EX20_B.EXE

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\PASCAL\EX20_C.PAS e \EXERC\CAP4\PASCAL\EX20_C.EXE

C/C++

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX20_A.CPP e \EXERC\CAP4\C++\EX20_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX20_B.CPP e \EXERC\CAP4\C++\EX20_B.EXE

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\C++\EX20_C.CPP e \EXERC\CAP4\C++\EX20_C.EXE

JAVA

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

EXERC\CAP4\JAVA\EX20_A.java e \EXERC\CAP4\JAVA\EX20_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX20_B.java e \EXERC\CAP4\JAVA\EX20_B.class

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\JAVA\EX20_C.java e \EXERC\CAP4\JAVA\EX20_C.class

21. Faça um programa que receba:

- o código do estado de origem da carga de um caminhão, supondo que a digitação do código do estado seja sempre válida, isto é, um número inteiro entre 1 e 5;
- o peso da carga do caminhão em toneladas;
- o código da carga, supondo que a digitação do código seja sempre válida, isto é, um número inteiro entre 10 e 40.

Tabelas:

CÓDIGO DO ESTADO	IMPOSTO	CÓDIGO DA CARGA	PREÇO POR QUILO
1	35%	10 a 20	100
2	25%	21 a 30	250
3	15%	31 a 40	340
4	5%		
5	Isento		

Calcule e mostre:

- o peso da carga do caminhão convertido em quilos;
- o preço da carga do caminhão;
- o valor do imposto, sabendo que o imposto é cobrado sobre o preço da carga do caminhão e depende do estado de origem;
- o valor total transportado pelo caminhão, preço da carga mais imposto.

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE cod_est, cod_carga, peso_toneladas NUMÉRICO
      peso_quilos, pre_carga, imposto, valor_total NUMÉRICO
LEIA cod_est, peso_toneladas, cod_carga
peso_quilos ← peso_toneladas * 1000
ESCREVA peso_quilos
SE cod_carga >= 10 E cod_carga <= 20
    ENTÃO pre_carga ← 100 * peso_quilos
SE cod_carga >= 21 E cod_carga <= 30
    ENTÃO pre_carga ← 250 * peso_quilos
SE cod_carga >= 31 E cod_carga <= 40
    ENTÃO pre_carga ← 340 * peso_quilos
ESCREVA pre_carga
SE cod_est = 1
    ENTÃO imposto ← 35/100 * pre_carga
SE cod_est = 2
    ENTÃO imposto ← 25/100 * pre_carga
SE cod_est = 3
    ENTÃO imposto ← 15/100 * pre_carga
SE cod_est = 4
    ENTÃO imposto ← 5/100 * pre_carga
SE cod_est = 5
    ENTÃO imposto ← 0
ESCREVA imposto
valor_total ← pre_carga + imposto
ESCREVA valor_total
FIM_ALGORITMO.

```

PASCAL

1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

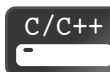
\EXERC\CAP4\PASCAL\EX21_A.PAS e \EXERC\CAP4\PASCAL\EX21_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\PASCAL\EX21_B.PAS e \EXERC\CAP4\PASCAL\EX21_B.EXE

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\PASCAL\EX21_C.PAS e \EXERC\CAP4\PASCAL\EX21_C.EXE



1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\C++\EX21_A.CPP e \EXERC\CAP4\C++\EX21_A.EXE

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\C++\EX21_B.CPP e \EXERC\CAP4\C++\EX21_B.EXE

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\C++\EX21_C.CPP e \EXERC\CAP4\C++\EX21_C.EXE



1ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL SIMPLES:

\EXERC\CAP4\JAVA\EX21_A.java e \EXERC\CAP4\JAVA\EX21_A.class

2ª SOLUÇÃO – UTILIZANDO ESTRUTURA CONDICIONAL COMPOSTA:

\EXERC\CAP4\JAVA\EX21_B.java e \EXERC\CAP4\JAVA\EX21_B.class

3ª SOLUÇÃO – UTILIZANDO ESTRUTURA SELETORA:

\EXERC\CAP4\JAVA\EX21_C.java e \EXERC\CAP4\JAVA\EX21_C.class

22. Faça um programa que receba o salário base e o tempo de serviço de um funcionário. Calcule e mostre:

- O imposto, conforme a tabela a seguir.

SALÁRIO BASE	% SOBRE O SALÁRIO BASE
< R\$ 200,00	isento
Entre R\$ 200,00 (inclusive) e R\$ 450,00 (inclusive)	3%
Entre R\$ 450,00 e R\$ 700,00	8%
>= R\$ 700,00	12%

- A gratificação, de acordo com a tabela a seguir.

SALÁRIO BASE	TEMPO DE SERVIÇO	GRATIFICAÇÃO
Superior a R\$ 500,00	Até 3 anos	20
	Mais de 3 anos	30
Até R\$ 500,00	Até 3 anos	23
	Entre 3 e 6 anos	35
	De 6 anos para cima	33

- O salário líquido, ou seja, salário base menos imposto mais gratificação.
- A categoria, que está na tabela a seguir.

SALÁRIO LÍQUIDO	CLASSIFICAÇÃO
Até R\$ 350,00	A
Entre R\$ 350,00 e R\$ 600,00	B
De R\$ 600,00 para cima	C

ALGORITMO SOLUÇÃO:

```
ALGORITMO
DECLARE sal_base, tempo, imposto, grat NUMÉRICO
        sal_liq NUMÉRICO
LEIA sal_base, tempo
SE sal_base < 200
```



```

ENTÃO imposto ← 0
SENÃO SE sal_base ≤ 450
    ENTÃO imposto ← 3/100 * sal_base
    SENÃO SE sal_base < 700
        ENTÃO imposto ← 8/100 * sal_base
        SENÃO imposto ← 12/100 * sal_base
ESCREVA imposto
SE sal_base > 500
ENTÃO INÍCIO
    SE tempo ≤ 3
        ENTÃO grat ← 20
        SENÃO grat ← 30
FIM
SENÃO INÍCIO
    SE tempo ≤ 3
        ENTÃO grat ← 23
        SENÃO SE tempo < 6
            ENTÃO grat ← 35
            SENÃO grat ← 33
FIM
ESCREVA grat
sal_liq ← sal_base - imposto + grat
ESCREVA sal_liq
SE sal_liq ≤ 350
    ENTÃO ESCREVA "Classificação A"
    SENÃO SE sal_liq < 600
        ENTÃO ESCREVA "Classificação B"
        SENÃO ESCREVA "Classificação C"
FIM_ALGORITMO.

```

PASCALSOLUÇÃO:

\EXERC\CAP4\PASCAL\EX22.PAS e \EXERC\CAP4\PASCAL\EX22.EXE

C/C++SOLUÇÃO:

\EXERC\CAP4\C++\EX22.CPP e \EXERC\CAP4\C++\EX22.EXE

JAVASOLUÇÃO:

\EXERC\CAP4\JAVA\EX22.java e \EXERC\CAP4\JAVA\EX22.class

23. Faça um programa que receba o valor do salário mínimo, o turno de trabalho (M — matutino; V — vespertino; ou N — noturno), a categoria (O — operário; G — gerente) e o número de horas trabalhadas no mês de um funcionário. Suponha a digitação apenas de dados válidos e, quando houver digitação de letras, utilize maiúsculas. Calcule e mostre:

- O coeficiente do salário, de acordo com a tabela a seguir.

TURNO DE TRABALHO	VALOR DO COEFICIENTE
M — Matutino	10% do salário mínimo
V — Vespertino	15% do salário mínimo
N — Noturno	12% do salário mínimo

- O valor do salário bruto, ou seja, o número de horas trabalhadas multiplicado pelo valor do coeficiente do salário.
- O imposto, de acordo com a tabela a seguir.

CATEGORIA	SALÁRIO BRUTO	IMPOSTO SOBRE O SALÁRIO BRUTO
O – Operário	≥ R\$ 300,00	5%
	< R\$ 300,00	3%
G – Gerente	≥ R\$ 400,00	6%
	< R\$ 400,00	4%

- A gratificação, de acordo com as regras a seguir.

Se o funcionário preencher **todos** os requisitos a seguir, sua gratificação será de R\$ 50,00; caso contrário, será de R\$ 30,00. Os requisitos são:

Turno: Noturno

Número de horas trabalhadas: Superior a 80 horas

- O auxílio alimentação, de acordo com as seguintes regras.

Se o funcionário preencher **algum** dos requisitos a seguir, seu auxílio alimentação será de um terço do seu salário bruto; caso contrário, será de metade do seu salário bruto. Os requisitos são:

Categoria: Operário

Coeficiente do salário: < = 25

- O salário líquido, ou seja, salário bruto menos imposto mais gratificação mais auxílio alimentação.
- A classificação, de acordo com a tabela a seguir:

SALÁRIO LÍQUIDO	MENSAGEM
Menor que R\$ 350,00	Mal remunerado
Entre R\$ 350,00 e R\$ 600,00	Normal
Maior que R\$ 600,00	Bem remunerado

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE sal_min, nht, coeficiente, sal_bruto NUMÉRICO
        imposto, grat, auxilio, sal_liq NUMÉRICO
        turno, categoria LITERAL
LEIA sal_min, turno, categoria, nht
SE turno = "M"
    ENTÃO coeficiente ← 10/100 * sal_min
SE turno = "V"
    ENTÃO coeficiente ← 15/100 * sal_min
SE turno = "N"
    ENTÃO coeficiente ← 12/100 * sal_min
ESCREVA coeficiente
sal_bruto ← nht * coeficiente
ESCREVA sal_bruto
SE categoria = "O"
    ENTÃO INÍCIO
        SE sal_bruto ≥ 300
            ENTÃO imposto ← 5/100 * sal_bruto
            SENÃO imposto ← 3/100 * sal_bruto
        FIM
    SENÃO INÍCIO
        SE sal_bruto ≥ 400
            ENTÃO imposto ← 6/100 * sal_bruto
            SENÃO imposto ← 4/100 * sal_bruto
        FIM
ESCREVA imposto
SE turno = "N" E nht > 80
    ENTÃO grat ← 50
    SENÃO grat ← 30
ESCREVA grat
SE categoria = "O" OU coeficiente ≤ 25

```

```

        ENTÃO auxilio ← 1/3 * sal_bruto
        SENÃO auxilio ← 1/2 * sal_bruto
    ESCRIVA auxilio
    sal_liq ← sal_bruto - imposto + grat + auxilio
    ESCRIVA sal_liq
    SE sal_liq < 350
        ENTÃO ESCRIVA "Mal Remunerado"
    SE sal_liq >= 350 E sal_liq <= 600
        ENTÃO ESCRIVA "Normal"
    SE sal_liq > 600
        ENTÃO ESCRIVA "Bem Remunerado"
    FIM_ALGORITMO.

```

PASCALSOLUÇÃO:

\EXERC\CAP4\PASCAL\EX23.PAS e \EXERC\CAP4\PASCAL\EX23.EXE

C/C++SOLUÇÃO:

\EXERC\CAP4\C++\EX23.CPP e \EXERC\CAP4\C++\EX23.EXE

JAVASOLUÇÃO:

\EXERC\CAP4\JAVA\EX23.java e \EXERC\CAP4\JAVA\EX23.class

- 24.** Faça um programa que receba o preço, o tipo (A — alimentação; L — limpeza; e V — vestuário) e a refrigeração (S — produto que necessita de refrigeração; e N — produto que não necessita de refrigeração) de um produto. Suponha que haverá apenas a digitação de dados válidos e, quando houver digitação de letras, utilize maiúsculas. Calcule e mostre:

- O valor adicional, de acordo com a tabela a seguir:

REFRIGERAÇÃO	TIPO	PREÇO	VALOR ADICIONAL
N	A	< R\$ 15,00	R\$ 2,00
		≥ R\$ 15,00	R\$ 5,00
	L	< R\$ 10,00	R\$ 1,50
		≥ R\$ 10,00	R\$ 2,50
	V	< R\$ 30,00	R\$ 3,00
		≥ R\$ 30,00	R\$ 2,50
S	A		R\$ 8,00
	L		R\$ 0,00
	V		R\$ 0,00

O valor do imposto, de acordo com a regra a seguir.

PREÇO	PERCENTUAL SOBRE O PREÇO
< R\$ 25,00	5%
≥ R\$ 25,00	8%

- O preço de custo, ou seja, preço mais imposto.
- O desconto, de acordo com a regra a seguir.

O produto que não preencher nenhum dos requisitos a seguir terá desconto de 3%, caso contrário, 0 (zero).

Os requisitos são:

Tipo: A

Refrigeração: S

- O novo preço, ou seja, preço de custo mais adicional menos desconto.
- A classificação, de acordo com a regra a seguir.

NOVO PREÇO	CLASSIFICAÇÃO
\leq R\$ 50,00	Barato
Entre R\$ 50,00 e R\$ 100,00	Normal
\geq R\$ 100,00	Caro

ALGORITMO Solução:

```

ALGORITMO
DECLARE pre, valor_adic, imposto NUMÉRICO
        pre_custo, desconto, novo_pre NUMÉRICO
        tipo, refrig LITERAL
LEIA pre, tipo, refrig
SE refrig = "N"
    ENTÃO INÍCIO
        SE tipo = "A"
            ENTÃO INÍCIO
                SE pre < 15
                    ENTÃO valor_adic ← 2
                    SENÃO valor_adic ← 5
                FIM
            SE tipo = "L"
                ENTÃO INÍCIO
                    SE pre < 10
                        ENTÃO valor_adic ← 1,50
                        SENÃO valor_adic ← 2,50
                    FIM
            SE tipo = "V"
                ENTÃO INÍCIO
                    SE pre < 30
                        ENTÃO valor_adic ← 3
                        SENÃO valor_adic ← 2,5
                    FIM
            FIM
        SENÃO INÍCIO
            SE tipo = "A"
                ENTÃO valor_adic ← 8
            SE tipo = "L"
                ENTÃO valor_adic ← 0
            SE tipo = "V"
                ENTÃO valor_adic ← 0
            FIM
        ESCREVA valor_adic
        SE pre < 25
            ENTÃO imposto ← 5/100 * pré
            SENÃO imposto ← 8/100 * pre
        ESCREVA imposto
        pre_custo ← pre + imposto
        ESCREVA pre_custo
        SE tipo ≠ "A" E refrig ≠ "S"
            ENTÃO desconto ← 3/100 * pre_custo
            SENÃO desconto ← 0
        ESCREVA desconto
        novo_pre ← pre_custo + valor_adic - desconto
        ESCREVA novo_pre
        SE novo_pre <= 50

```

```

    ENTÃO ESCRIVA "Barato"
    SENÃO SE novo_pre < 100
        ENTÃO ESCRIVA "Normal"
        SENÃO ESCRIVA "Caro"
FIM_ALGORITMO.

```

PASCAL SOLUÇÃO:

\EXERC\CAP4\PASCAL\EX24.PAS e \EXERC\CAP4\PASCAL\EX24.EXE

C/C++ SOLUÇÃO:

\EXERC\CAP4\C++\EX24.CPP e \EXERC\CAP4\C++\EX24.EXE

JAVA SOLUÇÃO:

\EXERC\CAP4\JAVA\EX24.java e \EXERC\CAP4\JAVA\EX24.class

- 25.** Faça um programa que receba a medida de um ângulo em graus. Calcule e mostre o quadrante em que se localiza esse ângulo. Considere os quadrantes da trigonometria e, para ângulos maiores que 360° ou menores que -360° , reduzi-los, mostrando também o número de voltas e o sentido da volta (horário ou anti-horário).

ALGORITMO SOLUÇÃO:

```

ALGORITMO
DECLARE angulo, voltas NUMÉRICO
LEIA angulo
SE angulo > 360 OU angulo < -360
    ENTÃO INÍCIO
        voltas ← parte inteira(angulo / 360)
        angulo ← RESTO(angulo / 360)
    FIM
    SENÃO voltas ← 0
SE angulo = 0 OU angulo = 90 OU angulo = 180
    OU angulo = 270 OU angulo = 360
    OU angulo = -90 OU angulo = -180
    OU angulo = -270 OU angulo = -360
    ENTÃO ESCRIVA "Está em cima de algum dos eixos"
SE (angulo > 0 E angulo < 90) OU (angulo < -270 E angulo > -360)
    ENTÃO ESCRIVA "1º Quadrante"
SE (angulo > 90 E angulo < 180) OU (angulo < -180 E angulo > -270)
    ENTÃO ESCRIVA "2º Quadrante"
SE (angulo > 180 E angulo < 270) OU (angulo < -90 E angulo > -180)
    ENTÃO ESCRIVA "3º Quadrante"
SE (angulo > 270 E angulo < 360) OU (angulo < 0 E angulo > -90)
    ENTÃO ESCRIVA "4º Quadrante"
ESCREVA voltas, " volta(s) no sentido "
SE angulo < 0
    ENTÃO ESCRIVA "horário"
    SENÃO ESCRIVA "anti-horário"
FIM_ALGORITMO.

```

PASCAL SOLUÇÃO:

\EXERC\CAP4\PASCAL\EX25.PAS e \EXERC\CAP4\PASCAL\EX25.EXE

C/C++ SOLUÇÃO:

\EXERC\CAP4\C++\EX25.CPP e \EXERC\CAP4\C++\EX25.EXE



SOLUÇÃO:

```
\EXERC\CAP4\JAVA\EX25.java e \EXERC\CAP4\JAVA\EX25.class
```

EXERCÍCIOS PROPOSTOS

1. Faça um programa que receba quatro notas de um aluno, calcule e mostre a média aritmética das notas e a mensagem de aprovado ou reprovado, considerando para aprovação média 7.
2. Faça um programa que receba duas notas, calcule e mostre a média aritmética e a mensagem que se encontra na tabela a seguir:

MÉDIA ARITMÉTICA		MENSAGEM	
0,0	● — ○	3,0	Reprovado
3,0	● — ○	7,0	Exame
7,0	● — ●	10,0	Aprovado

3. Faça um programa que receba dois números e mostre o menor.
4. Faça um programa que receba três números e mostre o maior.
5. Faça um programa que receba dois números e execute as operações listadas a seguir, de acordo com a escolha do usuário.

ESCOLHA DO USUÁRIO	OPERAÇÃO
1	Média entre os números digitados
2	Diferença do maior pelo menor
3	Produto entre os números digitados
4	Divisão do primeiro pelo segundo

Se a opção digitada for inválida, mostre uma mensagem de erro e termine a execução do programa. Lembre-se de que, na operação 4, o segundo número deve ser diferente de zero.

6. Faça um programa que receba dois números e execute uma das operações listadas a seguir, de acordo com a escolha do usuário. Se for digitada uma opção inválida, mostre mensagem de erro e termine a execução do programa. As opções são:
 - a) O primeiro número elevado ao segundo número.
 - b) Raiz quadrada de cada um dos números.
 - c) Raiz cúbica de cada um dos números.
7. Uma empresa decide dar um aumento de 30% aos funcionários com salários inferiores a R\$ 500,00. Faça um programa que receba o salário do funcionário e mostre o valor do salário reajustado ou uma mensagem, caso ele não tenha direito ao aumento.
8. Faça um programa para calcular e mostrar o salário reajustado de um funcionário. O percentual de aumento encontra-se na tabela a seguir.

SALÁRIO	PERCENTUAL DE AUMENTO
Até R\$ 300,00	35%
Acima de R\$ 300,00	15%

9. Um banco concederá um crédito especial aos seus clientes, de acordo com o saldo médio no último ano. Faça um programa que receba o saldo médio de um cliente e calcule o valor do crédito, de acordo com a tabela a seguir. Mostre o saldo médio e o valor do crédito.

SALDO MÉDIO	PERCENTUAL
Acima de R\$ 400,00	30% do saldo médio
R\$ 400,00 —●—○ R\$ 300,00	25% do saldo médio
R\$ 300,00 —●—○ R\$ 200,00	20% do saldo médio
Até R\$ 200,00	10% do saldo médio

- 10.** O preço ao consumidor de um carro novo é a soma do custo de fábrica com a porcentagem do distribuidor e dos impostos, ambos aplicados ao custo de fábrica. As porcentagens encontram-se na tabela a seguir. Faça um programa que receba o custo de fábrica de um carro e mostre o preço ao consumidor.

CUSTO DE FÁBRICA	% DO DISTRIBUIDOR	% dos IMPOSTOS
Até R\$ 12.000,00	5	isento
Entre R\$ 12.000,00 e R\$ 25.000,00	10	15
Acima de R\$ 25.000,00	15	20

- 11.** Faça um programa que receba o salário atual de um funcionário e, usando a tabela a seguir, calcule e mostre o valor do aumento e o novo salário.

SALÁRIO	PERCENTUAL DE AUMENTO
Até R\$ 300,00	15%
R\$ 300,00 —○—○ R\$ 600,00	10%
R\$ 600,00 —●—● R\$ 900,00	5%
Acima de R\$ 900,00	0%

- 12.** Faça um programa que receba o salário bruto de um funcionário e, usando a tabela a seguir, calcule e mostre o valor a receber. Sabe-se que este é composto pelo salário bruto acrescido de gratificação e descontado o imposto de 7% sobre o salário.

TABELA DAS GRATIFICAÇÕES	
SALÁRIO	GRATIFICAÇÃO
Até R\$ 350,00	R\$ 100,00
R\$ 350,00 —○—○ R\$ 600,00	R\$ 75,00
R\$ 600,00 —●—● R\$ 900,00	R\$ 50,00
Acima de R\$ 900,00	R\$ 35,00

- 13.** Faça um programa que receba o preço de um produto, calcule e mostre, de acordo com as tabelas a seguir, o novo preço e a classificação.

TABELA 1 — PERCENTUAL DE AUMENTO	
PREÇO	%
Até R\$ 50,00	5
Entre R\$ 50,00 e R\$ 100,00	10
Acima de R\$ 100,00	15

TABELA 2 — CLASSIFICAÇÕES	
NOVO PREÇO	CLASSIFICAÇÃO
Até R\$ 80,00	Barato
Entre R\$ 80,00 e R\$ 120,00 (inclusive)	Normal
Entre R\$ 120,00 e R\$ 200,00 (inclusive)	Caro
Maior que R\$ 200,00	Muito caro

- 14.** Faça um programa que receba o salário de um funcionário e, usando a tabela a seguir, calcule e mostre o novo salário.

FAIXA SALARIAL	% DE AUMENTO
Até R\$ 300,00	50%
R\$ 300,00 —●— R\$ 500,00	40%
R\$ 500,00 —●— R\$ 700,00	30%
R\$ 700,00 —●— R\$ 800,00	20%
R\$ 800,00 —●— R\$ 1.000,00	10%
Acima de R\$ 1.000,00	5%

- 15.** Uma agência bancária possui dois tipos de investimentos, conforme o quadro a seguir. Faça um programa que receba o tipo de investimento e seu valor, calcule e mostre o valor corrigido após um mês de investimento, de acordo com o tipo de investimento.

TIPO	DESCRIÇÃO	RENDIMENTO MENSAL
1	Poupança	3%
2	Fundos de renda fixa	4%

- 16.** Uma empresa decide aplicar descontos nos seus preços usando a tabela a seguir. Faça um programa que receba o preço atual de um produto e seu código, calcule e mostre o valor do desconto e o novo preço.

PREÇO ATUAL	% DE DESCONTO
Até R\$ 30,00	Sem desconto
Entre R\$ 30,00 e R\$ 100,00	10%
Acima de R\$ 100,00	15%

- 17.** Faça um programa que verifique a validade de uma senha fornecida pelo usuário. A senha é 4531. O programa deve mostrar uma mensagem de permissão de acesso ou não.
- 18.** Faça um programa que receba a idade de uma pessoa e mostre a mensagem de maioridade ou não.
- 19.** Faça um programa que receba a altura e o sexo de uma pessoa e calcule e mostre seu peso ideal, utilizando as seguintes fórmulas (onde h é a altura):
- para homens: $(72.7 * h) - 58$.
 - para mulheres: $(62.1 * h) - 44.7$.
- 20.** Faça um programa que receba a idade de um nadador e mostre sua categoria, usando as regras a seguir. Para idade inferior a 5, deverá mostrar mensagem.

CATEGORIA	IDADE
Infantil	5 a 7
Juvenil	8 a 10
Adolescente	11 a 15
Adulto	16 a 30
Sênior	Acima de 30

- 21.** Faça um programa que receba o preço de um produto e seu código de origem e mostre sua procedência. A procedência obedece à tabela a seguir.

CÓDIGO DE ORIGEM	PROCEDÊNCIA
1	Sul
2	Norte
3	Leste
4	Oeste
5 ou 6	Nordeste
7 ou 8 ou 9	Sudeste
10 a 20	Centro-oeste
21 a 30	Nordeste

- 22.** Faça um programa que receba a idade e o peso de uma pessoa. De acordo com a tabela a seguir, verifique e mostre em qual grupo de risco essa pessoa se encaixa.

IDADE	PESO		
	Até 60	Entre 60 e 90 (inclusive)	Acima de 90
Menores que 20	9	8	7
De 20 a 50	6	5	4
Maiores que 50	3	2	1

- 23.** Faça um programa que receba:

- o código do produto comprado; e
- a quantidade comprada do produto.

Calcule e mostre:

- o preço unitário do produto comprado, seguindo a Tabela I;
- o preço total da nota;
- o valor do desconto, seguindo a Tabela II e aplicado sobre o preço total da nota; e
- o preço final da nota depois do desconto.

TABELA I	
CÓDIGO	PREÇO
1 a 10	R\$ 10,00
11 a 20	R\$ 15,00
21 a 30	R\$ 20,00
31 a 40	R\$ 30,00

TABELA II	
PREÇO TOTAL DA NOTA	% DE DESCONTO
Até R\$ 250,00	5%
Entre R\$ 250,00 e R\$ 500,00	10%
Acima de R\$ 500,00	15%

- 24.** Faça um programa que receba o preço, a categoria (1 — limpeza; 2 — alimentação; ou 3 — vestuário) e a situação (R — produtos que necessitam de refrigeração; e N — produtos que não necessitam de refrigeração).

Calcule e mostre:

- O valor do aumento, usando as regras que se seguem.

PREÇO	CATEGORIA	PERCENTUAL DE AUMENTO
<= 25	1	5%
	2	8%
	3	10%
> 25	1	12%
	2	15%
	3	18%

- O valor do imposto, usando as regras a seguir.

O produto que preencher **pelo menos** um dos seguintes requisitos pagará imposto equivalente a 5% do preço; caso contrário, pagará 8%. Os requisitos são:

Categoria: 2

Situação: R

- O novo preço, ou seja, o preço mais aumento menos imposto.
- A classificação, usando as regras a seguir.

NOVO PREÇO	CLASSIFICAÇÃO
<= R\$ 50,00	Barato
Entre R\$ 50,00 e R\$ 120,00	Normal
>= R\$ 120,00	Caro

25. Uma empresa decidiu dar uma gratificação de Natal a seus funcionários, baseada no número de horas extras e no número de horas que o funcionário faltou ao trabalho. O valor do prêmio é obtido pela consulta à tabela que se segue, na qual:

$H = \text{número de horas extras} - (2/3 * (\text{número de horas falta}))$

H (MINUTOS)	PRÊMIO (R\$)
>= 2.400	500,00
1.800 ○————○ 2.400	400,00
1.200 ●————○ 1.800	300,00
600 ●————○ 1.200	200,00
< 600	100,00