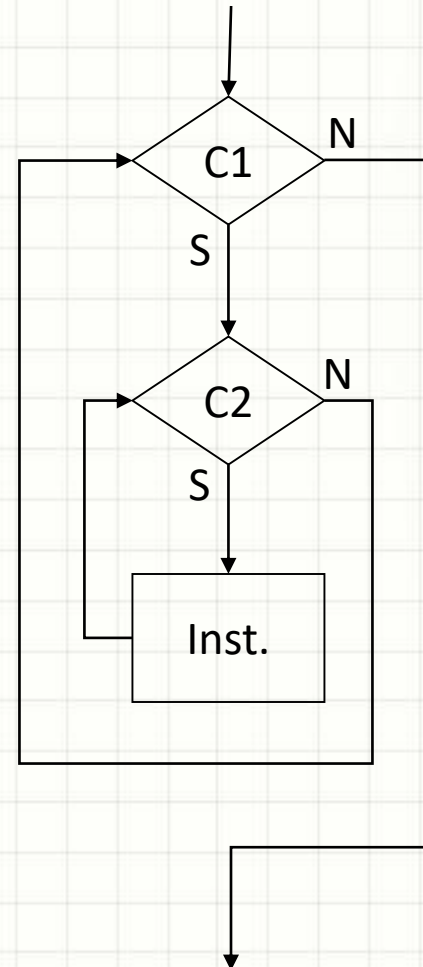


# Estruturas de Controle Encadeadas

- enquanto com enquanto

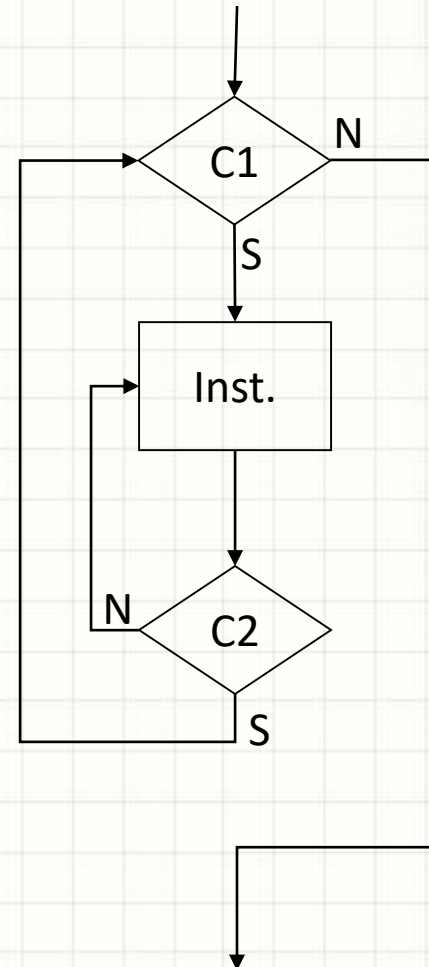
```
enquanto <condicao1> faça  
  enquanto <condicao2> faça  
    <instrucoes>  
  fim_enquanto  
fim_enquanto
```



# Estruturas de Controle Encadeadas

- enquanto com repita

```
enquanto <condicao1> faça  
  repita  
    <instrucoes>  
  até_que <condicao2>  
fim_enquanto
```



# Estruturas de Controle Encadeadas

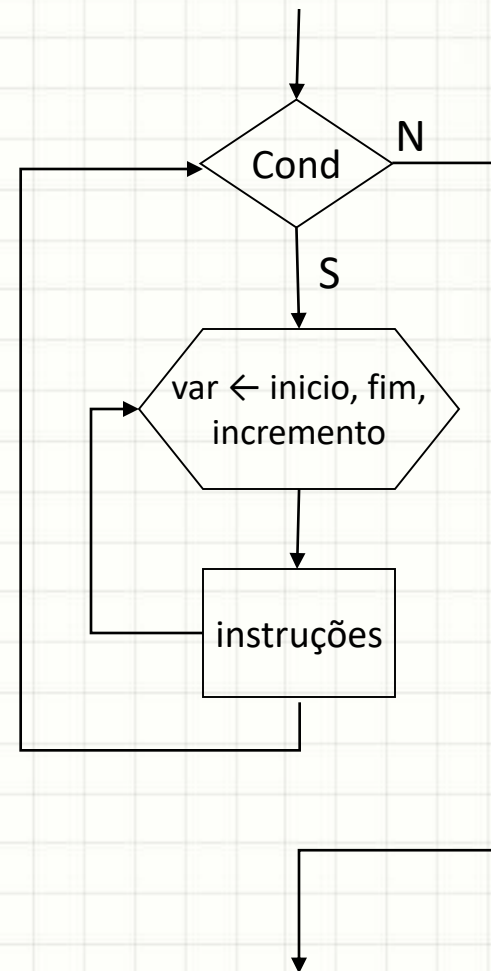
- enquanto com para

**enquanto** <condicao> **faça**

**para** <var> **de** <inicio> **até** <fim> **passo** <incr> **faça**  
    <instrucoes>

**fim\_para**

**fim\_enquanto**



# Estruturas de Controle Encadeadas

- repita com repita

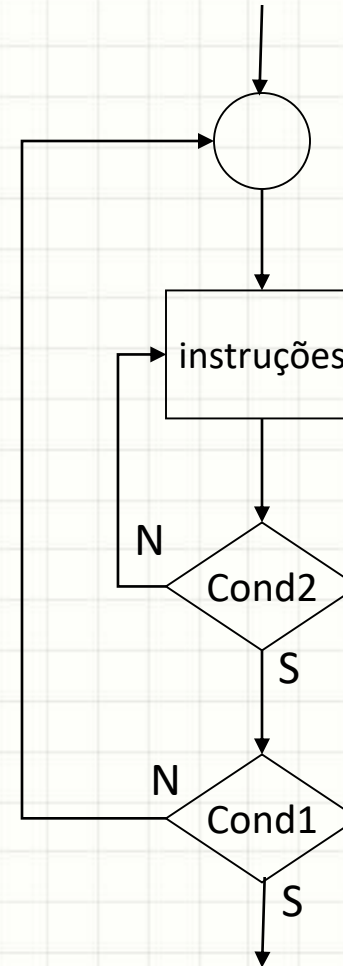
**repita**

**repita**

`<instrucoes>`

**até\_que** `<condicao2>`

**até\_que** `<condicao1>`



# Estruturas de Controle Encadeadas

- repita com enquanto

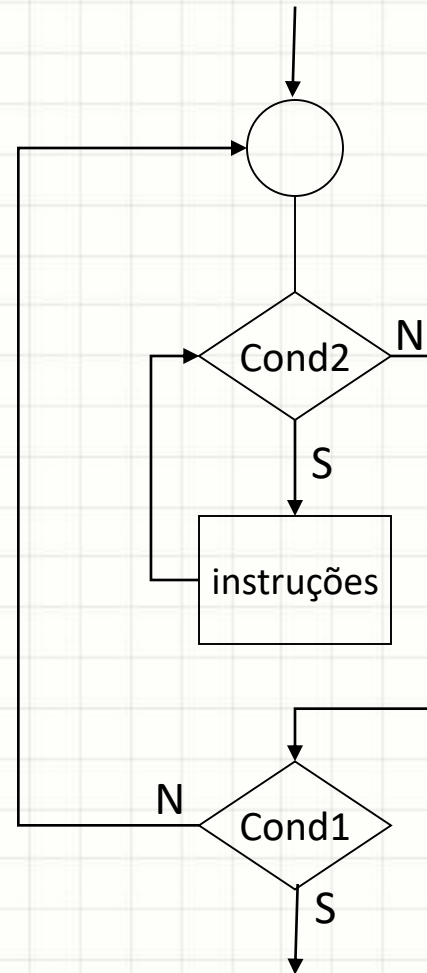
**repita**

**enquanto** <condicao2> **faça**

    <instrucoes>

**fim\_enquanto**

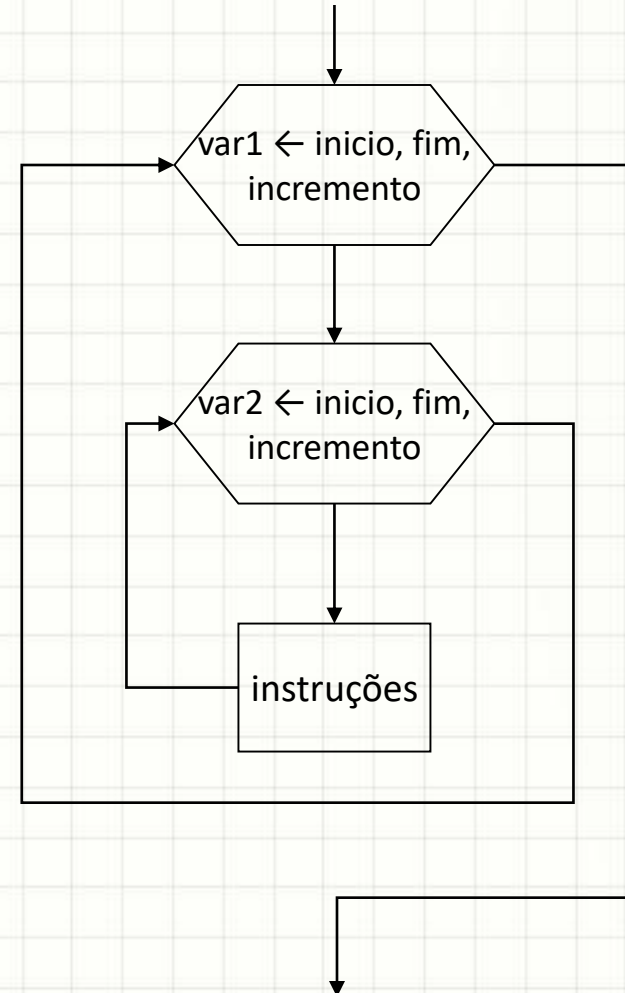
**até\_que** <condicao1>



# Estruturas de Controle Encadeadas

- para com para

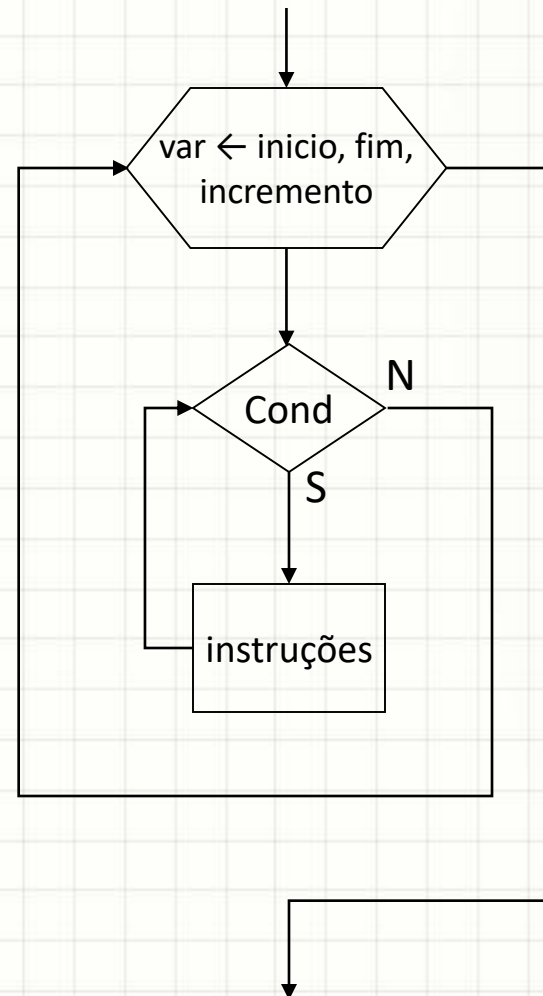
```
para <var1> de <inicio> até <fim> passo <incr> faça  
    para <var2> de <inicio> até <fim> passo <incr> faça  
        <instrucoes>  
    fim_para  
fim_para <condicao>
```



# Estruturas de Controle Encadeadas

- para com enquanto

```
para <var> de <inicio> até <fim> passo <incr> faça  
    enquanto <condicao> faça  
        <instrucoes>  
fim_enquanto  
fim_para
```

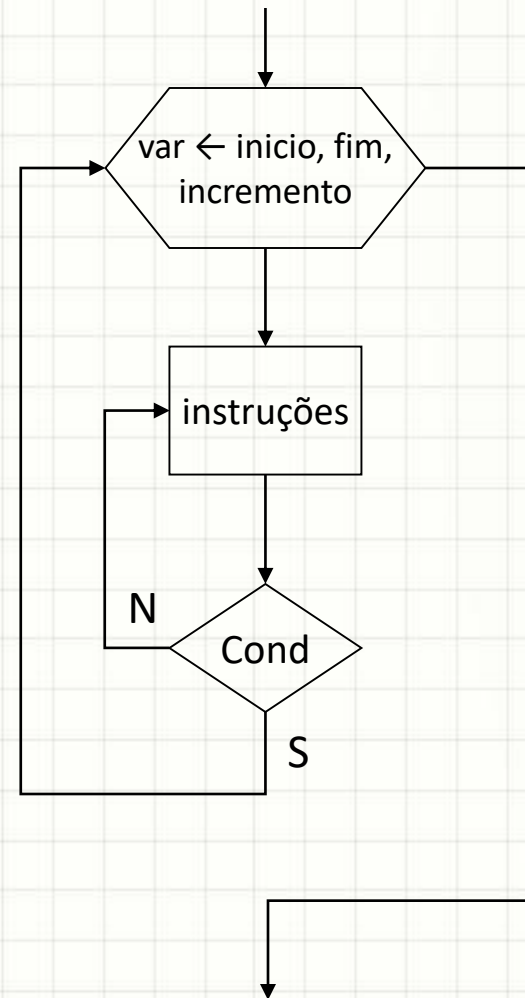




# Estruturas de Controle Encadeadas

- para com repita

```
para <var> de <inicio> até <fim> passo <incr> faça  
  repita  
    <instrucoes>  
  até_que <condicao>  
fim_para
```





# Exercícios

1. Assinale a alternativa que o valor final da variável X, após a execução do trecho de programa em Português Estruturado mostrado abaixo. Considere os seguintes valores iniciais para as variáveis: X=1;A=5.

```
enquanto (A < 9) faça  
    se (X < 6) então  
        X ← X * 2  
    fim_se  
    A ← A + 1  
fim_enquanto  
escreva X
```

# Exercícios

1. Assinale a alternativa que o valor final da variável X, após a execução do trecho de programa em Português Estruturado mostrado abaixo. Considere os seguintes valores iniciais para as variáveis: X=1;A=5.

```
enquanto (A < 9) faça  
    se (X < 6) então  
        X ← X * 2  
    fim_se  
    A ← A + 1  
fim_enquanto  
escreva X
```

Resposta: 8

# Exercícios

Indique a opção que contém o valor final da variável X, após a execução do trecho de programa em Português Estruturado mostrado abaixo. Considere os seguintes valores para as variáveis: A = 3; B = 2; C = 8; D = 7.

```
se .não. (A > 3) .e. .não. (B < 5) então
```

```
    X <- 10
```

```
senão
```

```
    se (A >= 2) .ou. (C <= 1) então
```

```
        X <- (A + D) / 2
```

```
    senão
```

```
        se (A = 2) .ou. (B < 7) então
```

```
            X <- (A + 2) * (B - 2)
```

```
        senão
```

```
            X <- ((A + C) / B) * (C+D)
```

```
        fim_se
```

```
    fim_se
```

```
fim_se
```

# Exercícios

Indique a opção que contém o valor final da variável X, após a execução do trecho de programa em Português Estruturado mostrado abaixo. Considere os seguintes valores para as variáveis: A = 3; B = 2; C = 8; D = 7.

```
se .não. (A > 3) .e. .não. (B < 5) então
    X <- 10
senão
    se (A >= 2) .ou. (C <= 1) então
        X <- (A + D) / 2
    senão
        se (A = 2) .ou. (B < 7) então
            X <- (A + 2) * (B - 2)
        senão
            X <- ((A + C) / B) * (C+D)
        fim_se
    fim_se
fim_se
```

Resposta: 5

# Exercícios

Indique a opção que contém o resultado do algoritmo abaixo.

**Início**

**Inteiro: I;**

**Para I de 4 até 10 passo 2 faça**

**Imprima (I);**

**Fim\_para;**

**Fim**

# Exercícios

Indique a opção que contém o resultado do algoritmo abaixo.

Início

Inteiro: I;

Para I de 4 até 10 passo 2 faça

Imprima (I);

Fim\_para;

Fim

Resposta: 4 , 6, 8 , 10

# Exercícios

Considere o algoritmo seguinte:

```
algoritmo "semnome"
var
    res, cont, x, n:inteiro

inicio
cont <- 6
res <- 0
x <- 2
n <- 4
enquanto (cont >= n) faça
    res <- res * x
    cont <- cont - 1
    escreva (res)
fimenquanto
Escreva (cont)
finalgoritmo
```

A saída na tela será?



# Exercícios

Considere o algoritmo seguinte:

```
algoritmo "semnome"
var
    res, cont, x, n:inteiro

inicio
cont <- 6
res <- 0
x <- 2
n <- 4
enquanto (cont >= n) faça
    res <- res * x
    cont <- cont - 1
    escreva (res)
fimenquanto
Escreva (cont)
finalgoritmo
```

A saída na tela será?

Resposta: 0, 0, 0, 3

# Matrizes de uma Dimensão ou Vetores

- Sua utilização mais comum está vinculada à criação de tabelas.
- Caracteriza-se por ser definida uma única variável dimensionada com um determinado tamanho.
- A dimensão de uma matriz é constituída por constantes inteiras e positivas.
- Os nomes dados às matrizes seguem as mesmas regras de nomes utilizados para indicar as variáveis simples.

# Matrizes de uma Dimensão ou Vetores (Utilização)

- Calcular e apresentar a média geral de uma turma de 8 alunos.
- A média a ser obtida deve ser a média geral das médias de cada aluno obtida durante o ano letivo.
- Desta forma será necessário somar todas as médias e dividi-las por 8.
- A tabela seguinte apresenta o número de alunos, suas notas bimestrais e respectivas médias anuais.
- É da média de cada aluno que será efetuado o cálculo da média da turma.

# Matrizes de uma Dimensão ou Vetores (Utilização)

Aluno	Nota 1	Nota 2	Nota 3	Nota 4	Média
1	4	6	5	3	4.5
2	6	7	5	8	6.5
3	9	8	9	6	8
4	3	5	4	2	3.5
5	4	6	6	8	6
6	7	7	7	7	7
7	8	7	6	5	6.5
8	6	7	2	9	6

# Matrizes de uma Dimensão ou Vetores

- Agora basta escrever um programa para efetuar o cálculo das 8 médias de cada aluno.
- Para representar a média do primeiro aluno será utilizada a avariável MD1, para o segundo MD2 e assim por diante. Então tem-se:

MD1 = 4.5

MD2 = 6.5

MD3 = 8.0

MD4 = 3.5

MD5 = 6.0

MD6 = 7.0

MD7 = 6.5

MD8 = 6.0

# Matrizes de uma Dimensão ou Vetores

- Com o conhecimento até este momento, seria então elaborado um programa que efetuaria a leitura de cada média, a soma dela e a divisão do valor da soma por 8, obtendo-se desta forma a média da turma, conforme exemplo abaixo em português estruturado:

Program media\_turma

Var

MD1, MD2, MD3, MD4, MD5, MD6, MD7, MD8 : real  
soma, media : real

Início

soma <- 0  
Leia MD1, MD2, MD3, MD4, MD5, MD6, MD7, MD8  
soma <- MD1 + MD2 + MD3 + MD4 + MD5 + MD6 + MD7 + MD8  
media <- soma / 8  
escreva media

fim



# Operações Básicas com Matrizes do tipo Vetor

- Uma matriz de uma dimensão ou vetor será, representada por seu nome e seu tamanho entre colchetes.
- Desta forma seria uma matriz MD[1..8], sendo seu nome MD, possuindo um tamanho de 1 a 8.
- Isto significa que poderão ser armazenados em MD até oito elementos.



# Operações Básicas com Matrizes do tipo Vetor

No caso do exemplo do cálculo da média dos 8 alunos, teremos então uma única variável indexada (a matriz) contendo todos os valores das 8 notas. Isto seria representado da seguinte forma:

MD[1] = 4.5

MD[2] = 6.5

MD[3] = 8.0

MD[4] = 3.5

MD[5] = 6.0

MD[6] = 7.0

MD[7] = 6.5

MD[8] = 6.0

Observe que o nome é um só. O que muda é a informação indicada dentro dos colchetes. A esta informação dá-se o nome de índice, sendo este o endereço em que o elemento está armazenado.

# Operações Básicas com Matrizes do tipo Vetor

No caso do exemplo do cálculo da média dos 8 alunos, como faríamos esse mesmo programa utilizando matrizes?

```
Program media_turma
```

```
Var
```

```
    MD : vetor[1..8] de real
```

```
    soma, media : real
```

```
    I : inteiro
```

```
Início
```

```
    soma <- 0
```

```
    para I de 1 até 8 passo 1 faça
```

```
        Leia MD[I]
```

```
        soma <- soma + MD[I]
```

```
    fim_para
```

```
    media <- soma / 8
```

```
    escreva media
```

```
fim
```

# Exemplo

O programa seguinte demonstra a utilização da rotina de pesquisa sequencial dentro de um contexto prático.

```
algoritmo "semnome"
```

```
var
```

```
nome : vetor[1..10] de caracter
```

```
l : inteiro
```

```
pesq, resp : caracter
```

```
acha : logico
```

```
Inicio
```

```
para l de 1 ate 10 passo 1 faca
```

```
    Leia (nome[l])
```

```
fimpara
```

```
    resp <- "Sim"
```

```
Enquanto (resp = "Sim") faca
```

```
    escreva ("Entre com o Nome: ")
```

```
    leia (pesq)
```

```
    l <- 1
```

```
    acha <- Falso
```

```
    enquanto (l <= 10) e (acha = Falso) faca
```

```
        se (pesq = nome[l]) entao
```

```
            acha <- Verdadeiro
```

```
        senao
```

```
            l <- l + 1
```

```
        fimse
```

```
    fimenquanto
```

```
    se (acha = Verdadeiro) entao
```

```
        escreva (pesq, " foi localizado na posição ", l)
```

```
    senao
```

```
        escreva pesq, " não foi encontrado"
```

```
    fimse
```

```
    escreva ("", deseja continuar?)
```

```
    leia (resp)
```

```
Fimenquanto
```

```
fimalgoritmo
```

# Exercícios

Criar um programa que efetue a leitura dos nomes de 20 pessoas e em seguida apresentá-los na mesma ordem em que foram informados.

```
Program lista_nome
Var
    nome : conjunto[1..20] de caracter
    I : inteiro
Início
    para I de 1 até 20 passo 1 faça
        Leia nome[I]
    fim_para
    para I de 1 até 20 passo 1 faça
        escreva nome[I]
    fim_para
fim
```

# Matrizes com mais de uma Dimensão

- Anteriormente, houve contato com o uso de uma única variável indexada com apenas uma dimensão (uma coluna e várias linhas).
- Agora serão apresentadas tabelas com mais colunas, sendo assim, teremos variáveis no sentido horizontal e vertical

# Matrizes com mais de uma Dimensão

- Em matrizes com mais de uma dimensão os seus elementos são também manipulados de forma individualizada, sendo a referência feita sempre por meio de dois índices: o primeiro para indicar a linha e o segundo para indicar a coluna.
- Exemplo:
- Tabela[2,3]



# Matrizes com mais de uma Dimensão

- Uma matriz de duas dimensões é representada por seu nome e seu tamanho (dimensão) entre colchetes. Desta forma a matriz ***Tabela[1..8,1..5]***, tem seu nome ***Tabela***, possui um tamanho de 8 linhas (de 1 a 8) e 5 colunas (de 1 a 5).
- Isso significa que pode ser armazenado em ***Tabela*** até 40 elementos
- Sintaxe:

Variável : conjunto[<dimensão1>,<dimensão2>] de <tipo de dado>



# Matrizes com mais de uma Dimensão

## Exemplo leitura

Program elementos

Var

notas : conjunto[1..8,1..5] de real

I, J : inteiro

Início

para I de 1 até 8 passo 1 faça

para J de 1 até 5 passo 1 faça

Leia notas[I,J]

fim\_para

fim\_para

fim

# Matrizes com mais de uma Dimensão

## Exemplo escrita

Program elementos

Var

notas : conjunto[1..8,1..5] de real

I, J : inteiro

Início

para I de 1 até 8 passo 1 faça

para J de 1 até 5 passo 1 faça

escreva notas[I,J]

fim\_para

fim\_para

fim

# Exercícios

Criar um programa que primeiro efetue a leitura dos nomes de 6 pessoas e após a leitura de todos os nomes, apresentá-los na tela. Utilizando uma matriz com duas dimensões.

Programa nomes

Var

nomes : conjunto[1..3,1..2] de caracter  
I, J : inteiro

Início

para I de 1 até 3 passo 1 faça  
    para J de 1 até 2 passo 1 faça  
        leia nomes[I,J]

    fim\_para

fim\_para

para I de 1 até 3 passo 1 faça  
    para J de 1 até 2 passo 1 faça  
        escreva nomes[I,J]

    fim\_para

fim\_para

fim

# Exercícios

O que será impresso após a execução do algoritmo abaixo?

```
algoritmo "vetor"  
var  
    i: inteiro  
    num: vetor[1..10] de inteiro  
inicio  
    para i de 1 ate 10 passo 1 faca  
        num[i] <- i + 10  
    fimpara  
    para i de 1 ate 10 passo 1 faca  
        escreva(num[i])  
        escreva(",")  
    fimpara  
fimalgoritmo
```

**11, 12, 13, 14, 15, 16, 17, 18, 19, 20**

# Exercícios

O que será impresso após a execução do algoritmo abaixo?

```
algoritmo "matriz"
var
  i, j: inteiro
  num: vetor[0..2, 0..2] de inteiro
inicio
  para i de 0 ate 2 passo 1 faca
    para j de 0 ate 2 passo 1 faca
      num[i, j] <- i + j
    fimpara
  fimpara
  escreval (" -----")
  para i de 0 ate 2 passo 1 faca
    para j de 0 ate 2 passo 1 faca
      escreva(" |", num[i, j])
    fimpara
    escreval (" |")
    escreval (" -----")
  fimpara
finalgoritmo
```

```
-----
| 0 | 1 | 2 |
-----
| 1 | 2 | 3 |
-----
| 2 | 3 | 4 |
```

# Exercícios de revisão

Um termômetro mede a temperatura em graus Fahrenheit. Crie um programa completo que receba um valor em Fahrenheit e informe a temperatura em graus Celsius. Ao final, solicite se deseja fazer uma nova conversão.

As temperaturas, em graus Celsius ( $^{\circ}\text{C}$ ) e em graus Fahrenheit ( $^{\circ}\text{F}$ ), são dados do tipo real.

A fórmula para conversão de Fahrenheit para Celsius é:  $^{\circ}\text{C} = 5$  dividido por 9, multiplicado pelo valor em Fahrenheit subtraído de 32.



# Exercícios de revisão

Dado o tipo do funcionário (Horista ou Mensalista), o salário por hora (horistas) ou por mês (mensalistas) e, o número de horas trabalhadas (horistas) ou o valor de faltas (mensalistas), calcular e mostrar o valor do salário bruto.

```
def calcular_salario_bruto(tipo_funcionario, salario_base, horas_trabalhadas, faltas):
    """
    Calcula o salário bruto com base no tipo de funcionário, salário base, horas trabalhadas ou faltas.
    """
    if tipo_funcionario == "Horista":
        salario_bruto = salario_base * horas_trabalhadas
    elif tipo_funcionario == "Mensalista":
        salario_bruto = salario_base - (faltas * 100)
    else:
        raise ValueError("Tipo de funcionário inválido")
    return salario_bruto
```



# Exercícios de revisão

Crie um programa que solicite 3 números aleatórios e depois exiba esses números em ordem crescente.

```
from random import randint

a = randint(0, 100)
b = randint(0, 100)
c = randint(0, 100)

print(a)
print(b)
print(c)

if a < b and a < c:
    print(a, b, c)
elif b < a and b < c:
    print(b, a, c)
elif c < a and c < b:
    print(c, a, b)
else:
    print(a, b, c)
```

## Exercícios de revisão

Crie um programa que leia dois números, efetuar as 4 operações matemáticas e mostrar os resultados

## Exercícios de revisão

João tem 1,50 metros e cresce 2 centímetros por ano, enquanto Maria tem 1,10 metros e cresce 3 centímetros por ano. Construa um algoritmo que calcule e imprima quantos anos serão necessários para que Maria seja maior que João.

# Exercícios de revisão

Faça um algoritmo que pergunte ao Usuário quantos números deseja somar. Em seguida, leia a quantidade informada de números e apresentar o valor da soma, quantos números são maiores que 7 e quantos números são maiores que 9.

```
1 1. Faça um algoritmo que pergunte ao
2 2. Usuário quantos números deseja somar.
3 3. Em seguida, leia a quantidade informada
4 4. de números e apresentar o valor da soma,
5 5. quantos números são maiores que 7 e
6 6. quantos números são maiores que 9.
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```