



# Estruturas de dados homogêneas

---

“Três classes de pessoas são infelizes:  
as que não sabem e não perguntam,  
as que sabem e não ensinam,  
as que ensinam e não fazem.”

V.Beda

# Dados de um aluno

```

inteiro idade <- 24
caracter sexo <- 'm'
real peso <- 74.32
real altura <- 1.75
inteiro nota1 <- 10
inteiro nota2 <- 5
inteiro nota3 <- 15
inteiro nota4 <- 17
inteiro nota5 <- 16
inteiro nota6 <- 3
inteiro nota7 <- 14
    
```

## Calcular

- Media das notas
- Nota máxima
- Numero de negativas

nome variavel	
endereço	valor

1 byte

	0	1	2	3	4	5	6	7	8
a	a:0	a:1	a:2	a:3	a:4	a:5	a:6	a:7	a:8
b	b:0	b:1	b:2	b:3	b:4	b:5	b:6	b:7	b:8
c	c:0	idade 24		c:3	sexo m'		c:6	c:7	c:8
d	Peso 74,32				altura 1,75				d:8
e	nota1 10	nota2 5	nota3 15	nota4 17	nota5 16	nota6 3	nota7 14	e:7	e:8
f	f:0	f:1	f:2	f:3	f:4	f:5	f:6	f:7	f:8
g	g:0	g:1	g:2	g:3	g:4	g:5	g:6	g:7	g:8

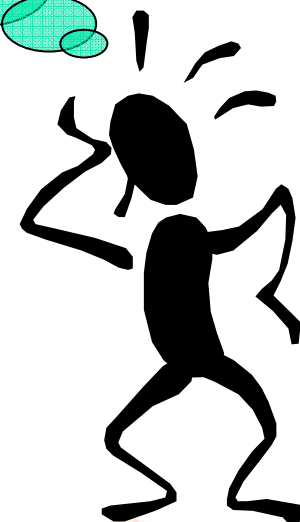
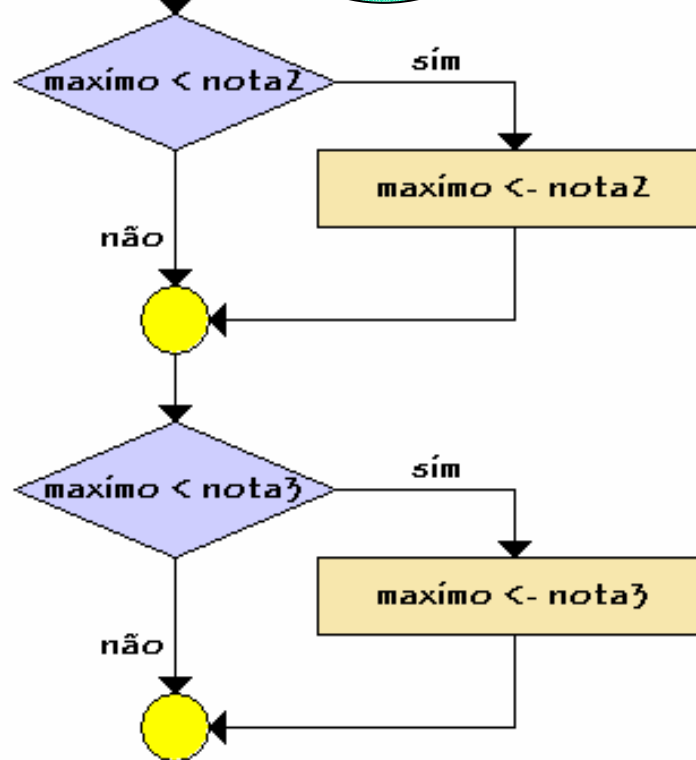
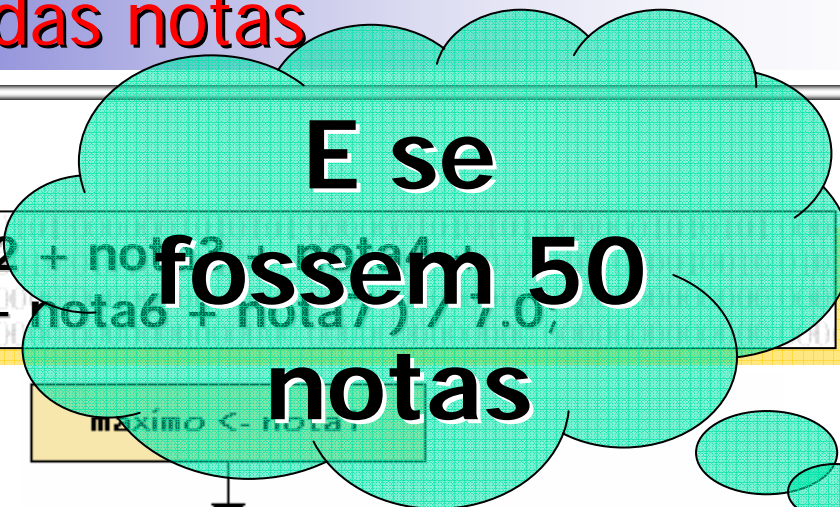
# Calcular a média das notas

## Media

```
media <- ( nota1 + nota2 + nota3 + nota4 + nota5 + nota6 + nota7 ) / 7.0;
```

## Nota máxima

```
inteiro nota_maxima <- nota1;
se nota2 > nota_maxima então
    nota_maxima = nota2;
Fimse
se nota3 > nota_maxima então
    nota_maxima = nota3;
Fimse
se nota4 > nota_maxima então
    nota_maxima = nota4;
Fimse
se nota5 > nota_maxima então
    nota_maxima = nota5;
Fimse
se nota6 > nota_maxima então
    nota_maxima = nota6;
Fimse
se nota7 > nota_maxima então
    nota_maxima = nota7;
Fimse
```



## Estruturas homogêneas de dados (arrays)

- Usam-se **arrays** para representar variáveis que são
  - Elementos do **mesmo tipo**
  - **Relacionados** entre si
  - Sobre os quais pretendemos fazer **operações semelhantes**
  - Que pretendemos representar por um **nome comum**
- Cada componente de um **array** é uma **variável** individual
  - Possui um **nome** (derivado do nome do array)
  - Possui um **tipo** (o do tipo base do array)
  - Possui um **valor** (manipulado como qualquer variável)
- Os **arrays** podem ser:
  - Unidimensionais (chamados de **vectores**)
  - Multidimensionais (chamados de **matrizes**)



# Vectores

- Declaração
  - tipo nome\_variável [numero de elementos]
    - inteiro nota[7]
- Acesso
  - Indexado pelo operador [ ]
    - nota[2]
    - Index <- 2
    - Nota [ index ]
- Leitura e escrita
  - Uma variável de cada vez

	0	1	2	3	4	5	6	7	8
<b>c</b>		<b>idade</b>			<b>sexo</b>				
	c:0	c:1	24	c:3	c:4	m'	c:5	c:6	c:7
<b>d</b>	<b>Peso</b>				<b>altura</b>				
	d:0	74,32		d:4	1,75		d:8		
<b>e</b>	<b>nota[0]</b>	<b>nota[1]</b>	<b>nota[2]</b>	<b>nota[3]</b>	<b>nota[4]</b>	<b>nota[5]</b>	<b>nota[6]</b>		
	e:0	e:1	e:2	e:3	e:4	e:5	3	e:6	e:7
	<b>nota</b>								

# Ler um vector

## Ler Vector

9

5

2

1

3

5

6

8

**inicio**

**inteiro nota [ 7 ]**

**inteiro index <- 0**

**enquanto index <= 6 faz**

**ler nota [ index ]**

**index <- index + 1**

**fimenquanto**

**fim**

## Ler Vector

**inicio**

**inteiro nota [ 7 ]**

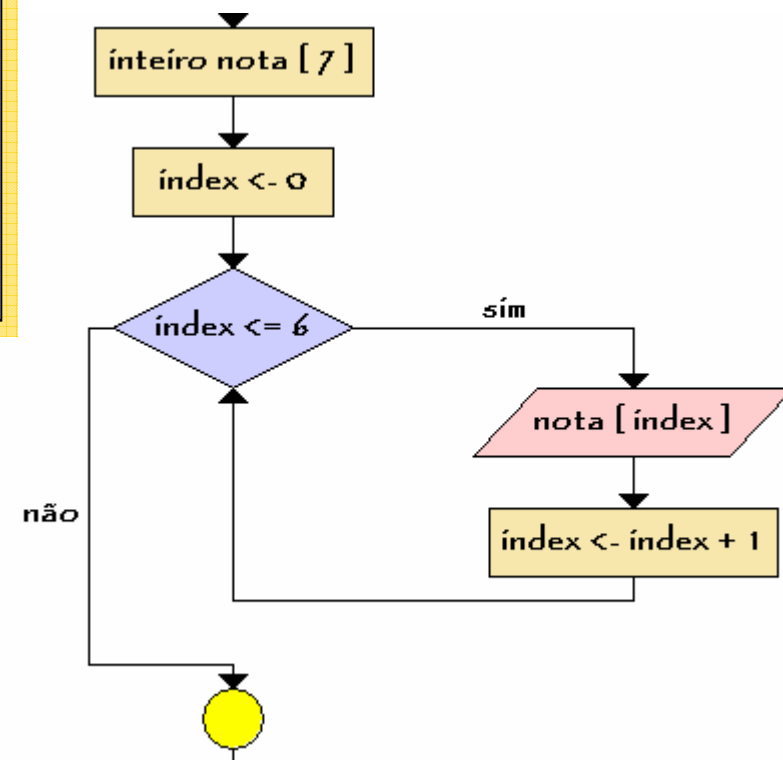
**inteiro index**

**para index de 0 ate 6**

**ler nota[index]**

**proximo**

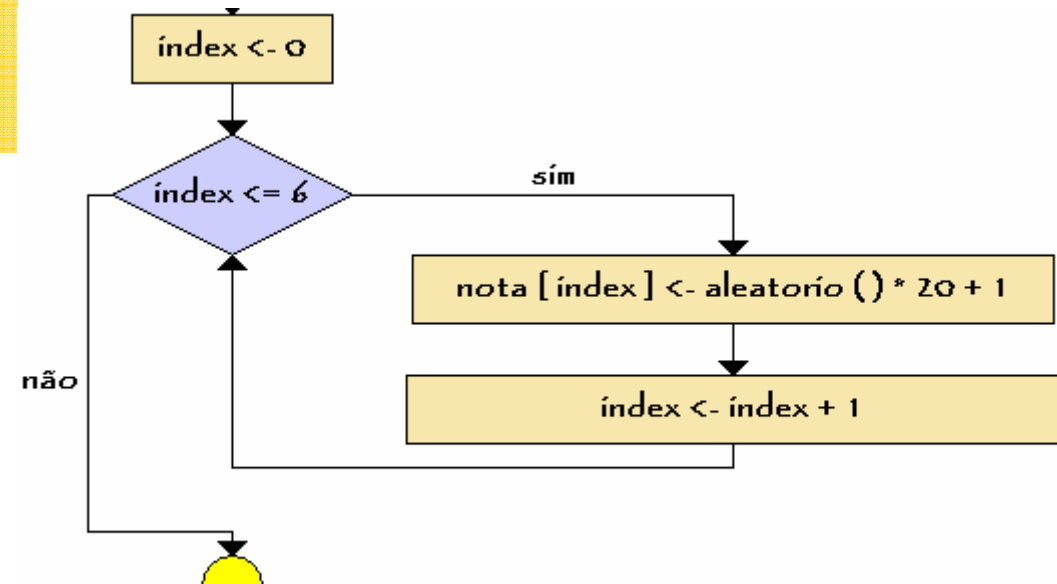
**fim**



# Inicializar um vector

Construa um programa que Inicialize o vector com notas aleatórias

9	5	2	1	3	5	6	8
---	---	---	---	---	---	---	---

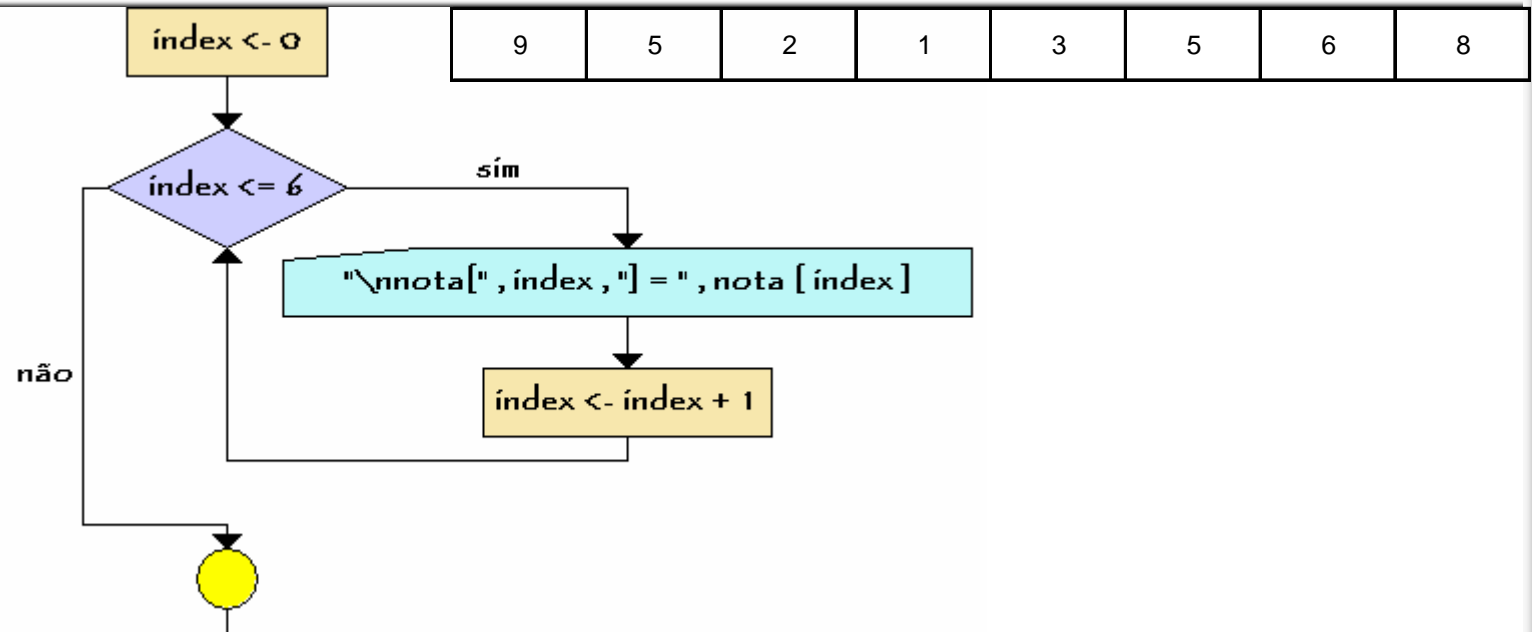


**inicializar**

```

para index de 0 ate 6
  nota[index] <- aleatorio() * 20 + 1
proximo
  
```

# escrever um vector



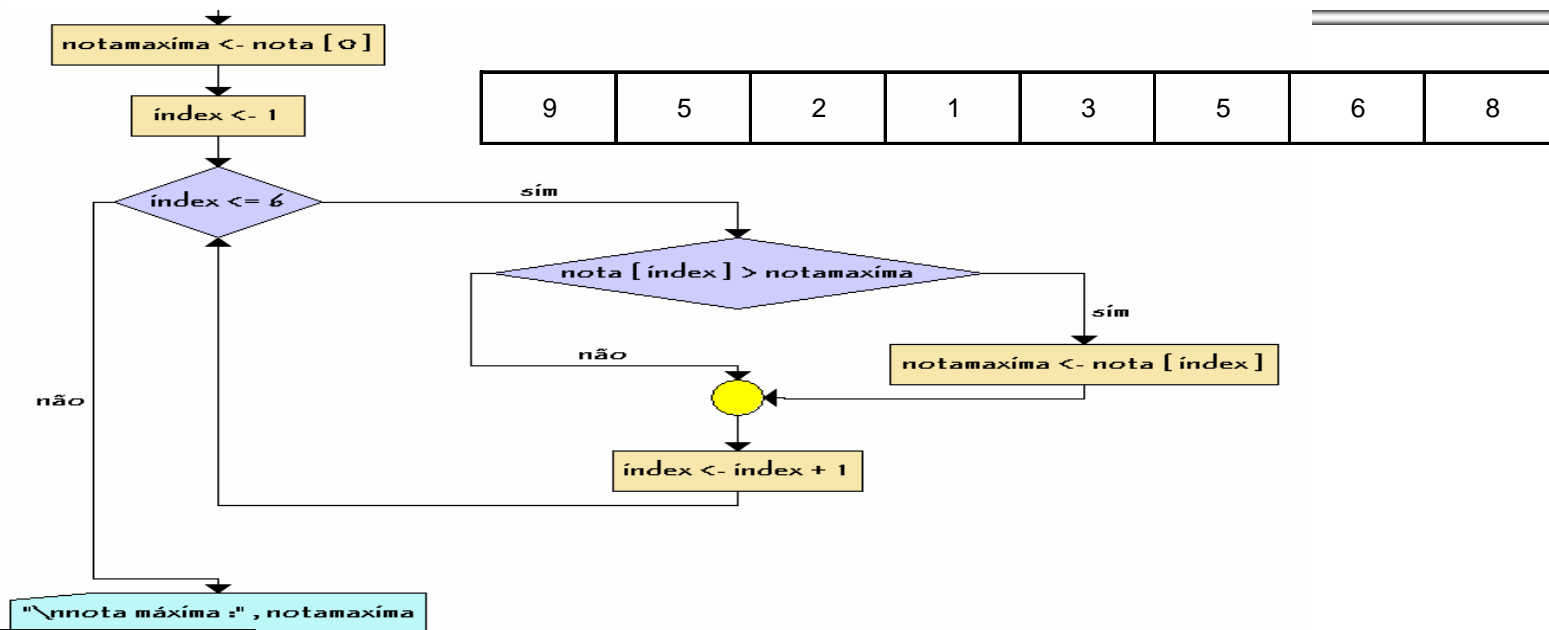
**escrever**

para index de 0 ate 6  
 escrever `"\nnota[",index,"] = ",nota[index]`  
 proximo





# Máximo de um vector



## Calcular o máximo

```
inteiro notamaxima <- nota [ 0 ]
```

```
para index de 1 ate 6
```

```
se nota [ index ] > notamaxima entao
```

```
notamaxima <- nota [ index ]
```

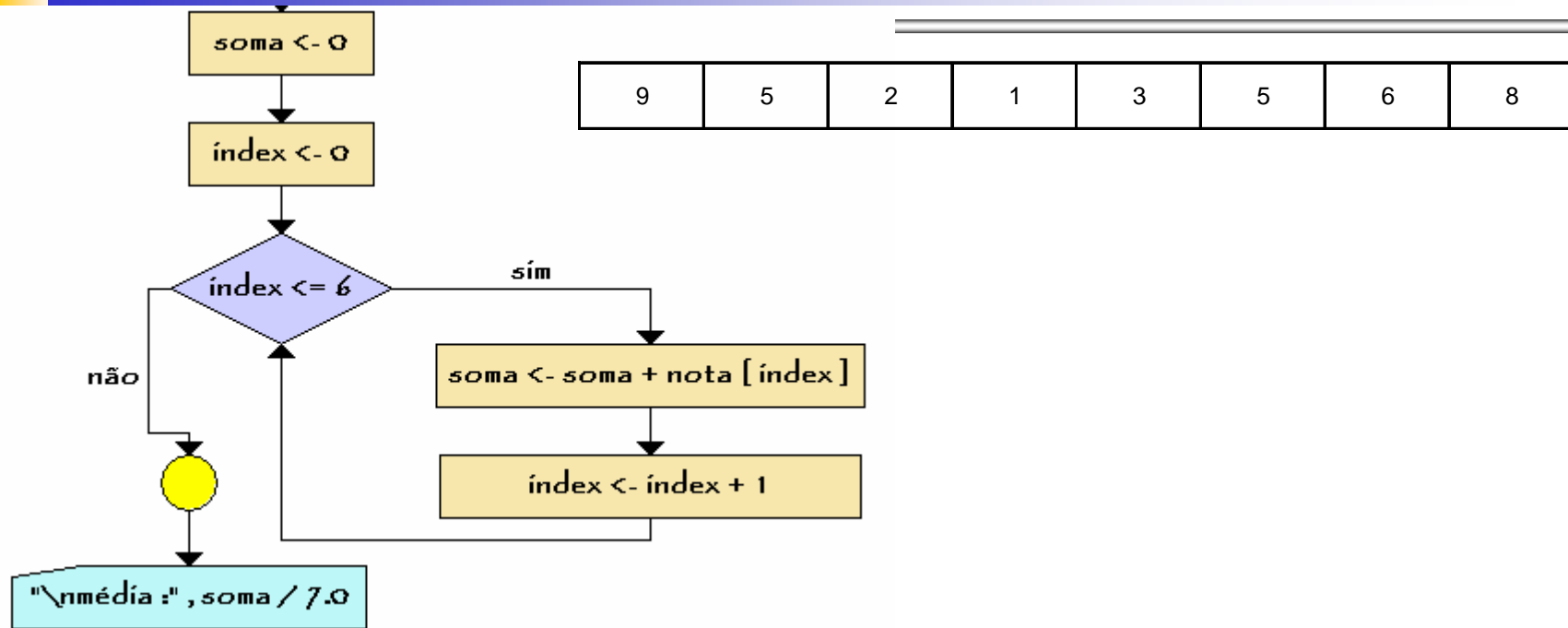
```
fimse
```

```
proximo
```

```
escrever "\nnota máxima :", notaMaxima
```



# Média de um vector



## Calcular a média

```
real soma <- 0
```

```
para index de 0 ate 6
```

```
  soma <- soma + nota [ index ]
```

```
proximo
```

```
escrever "\nmédia :", soma / 7.0
```



## Exercício

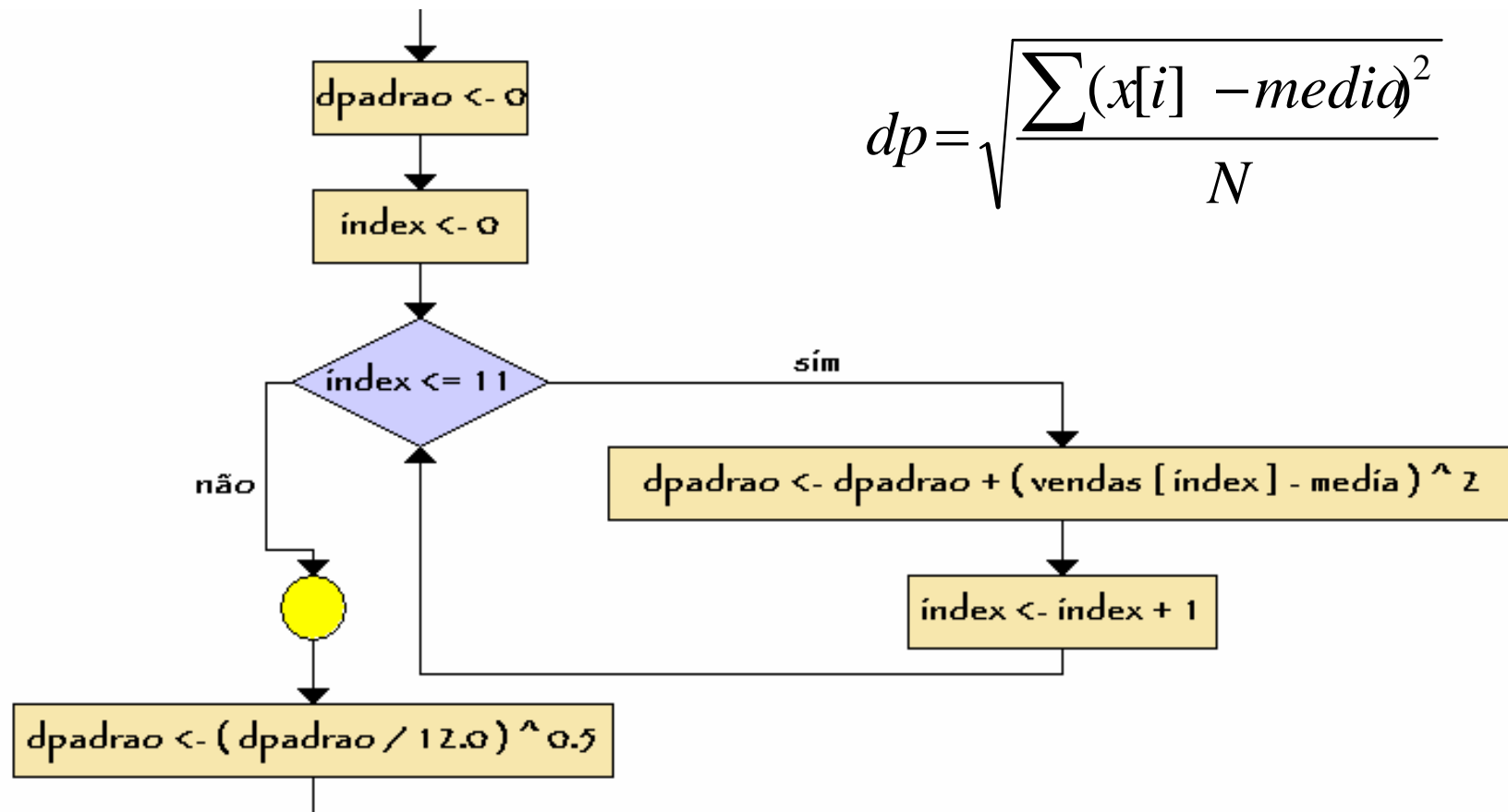
- Pretende-se uma aplicação que avalie a regularidade de vendas dos trabalhadores de uma empresa.
- Escreva um programa que realize a leitura as comissões pagas a um trabalhador durante um ano, calcule a média de comissões pagas e o seu desvio padrão.

$$dp = \sqrt{\frac{\sum (x[i] - \text{media})^2}{N}}$$

9	5	2	1	3	5	6	8
---	---	---	---	---	---	---	---



# Desvio padrão



$$dp = \sqrt{\frac{\sum (x[i] - media)^2}{N}}$$





# Matrizes

---



Várias dimensões

## Matrizes (vectors multidimensionais)

- Não existe limite para o número de dimensões de um vector
- Sintaxe
  - tipo nome\_variavel[dim\_1][dim\_2]...[dim\_n]
- Exemplos
  - inteiro notas[20][3]
  - character galo[3][3]

0		
	X	
0		X



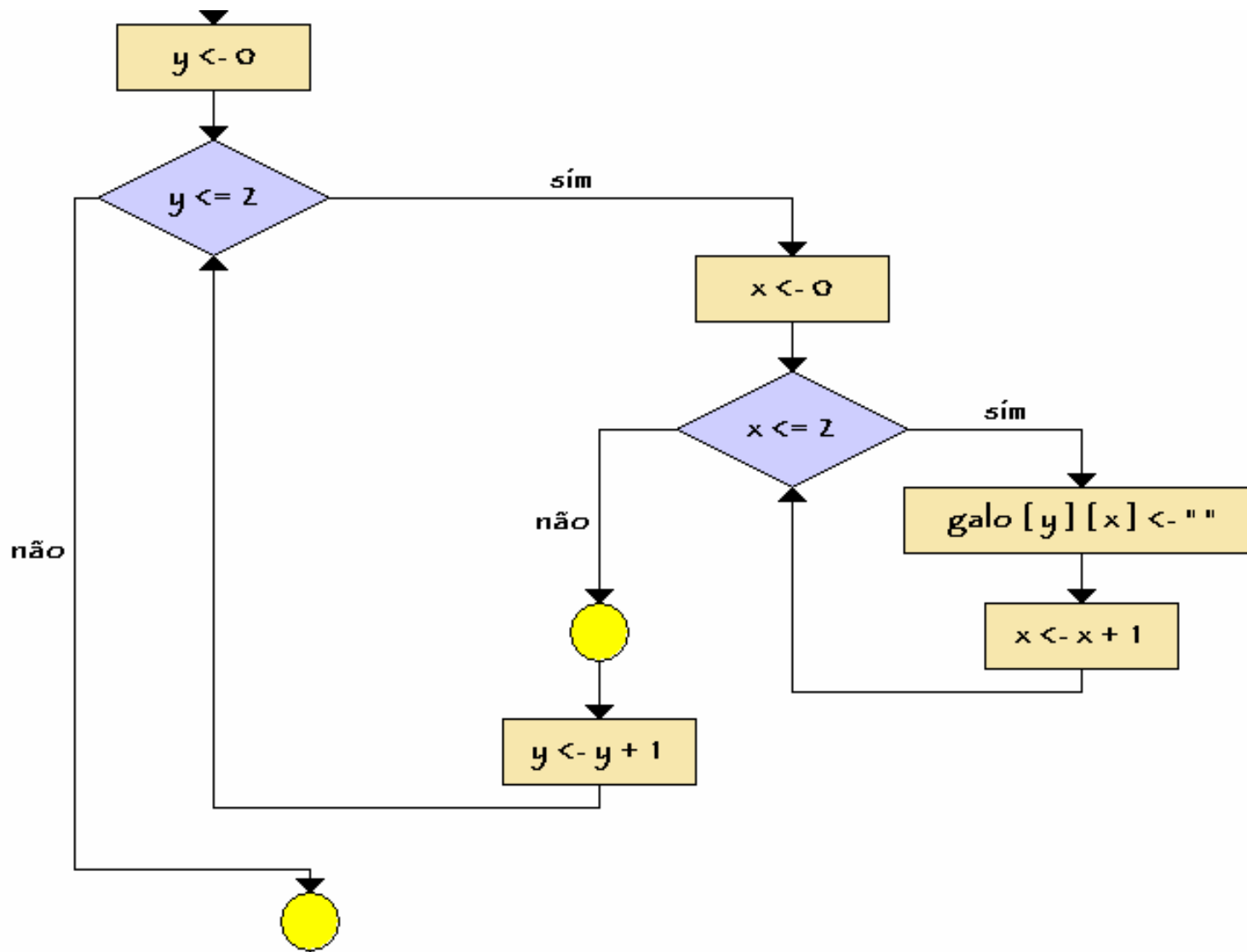
# Jogo do Galo

- Construa uma aplicação que permita jogar o jogo do galo
  - Declarar o Jogo
    - Caracter galo[3][3]
  - Inicializar o jogo
  - Apresentar o jogo
  - Jogar
    - Mudar de marca
    - Pedir a linha
    - Pedir a coluna

O		
	X	
O		X

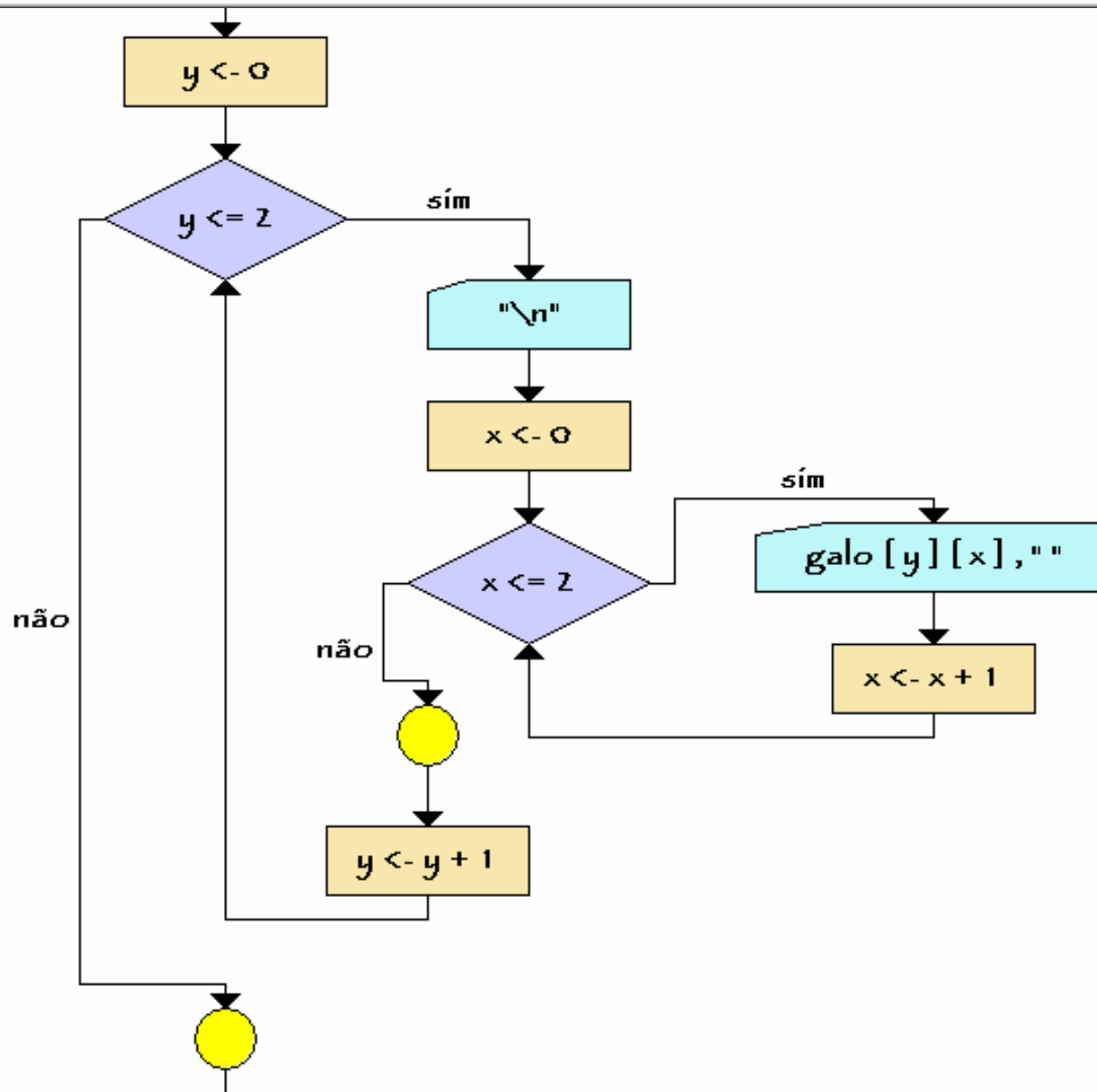


# Inicializar o jogo

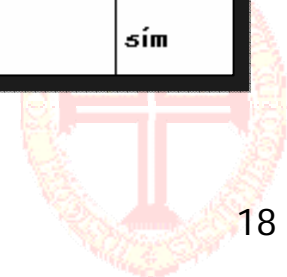
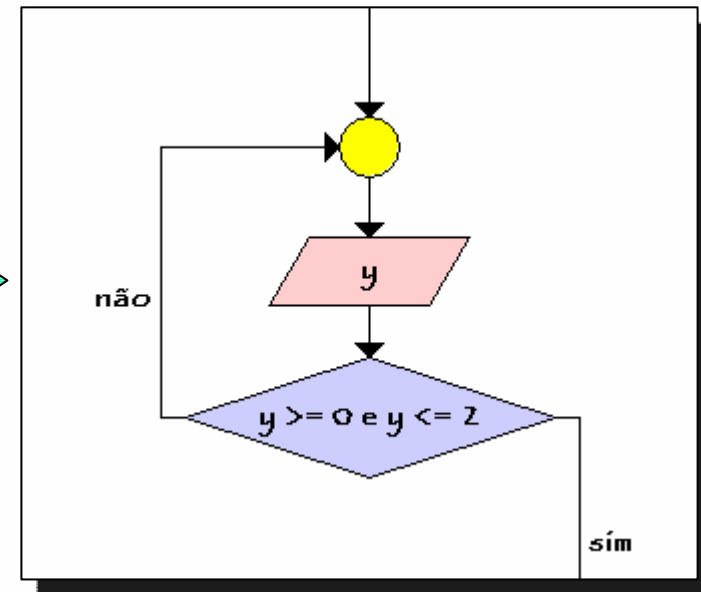
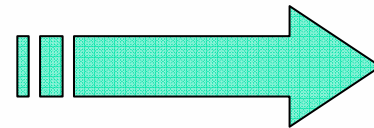
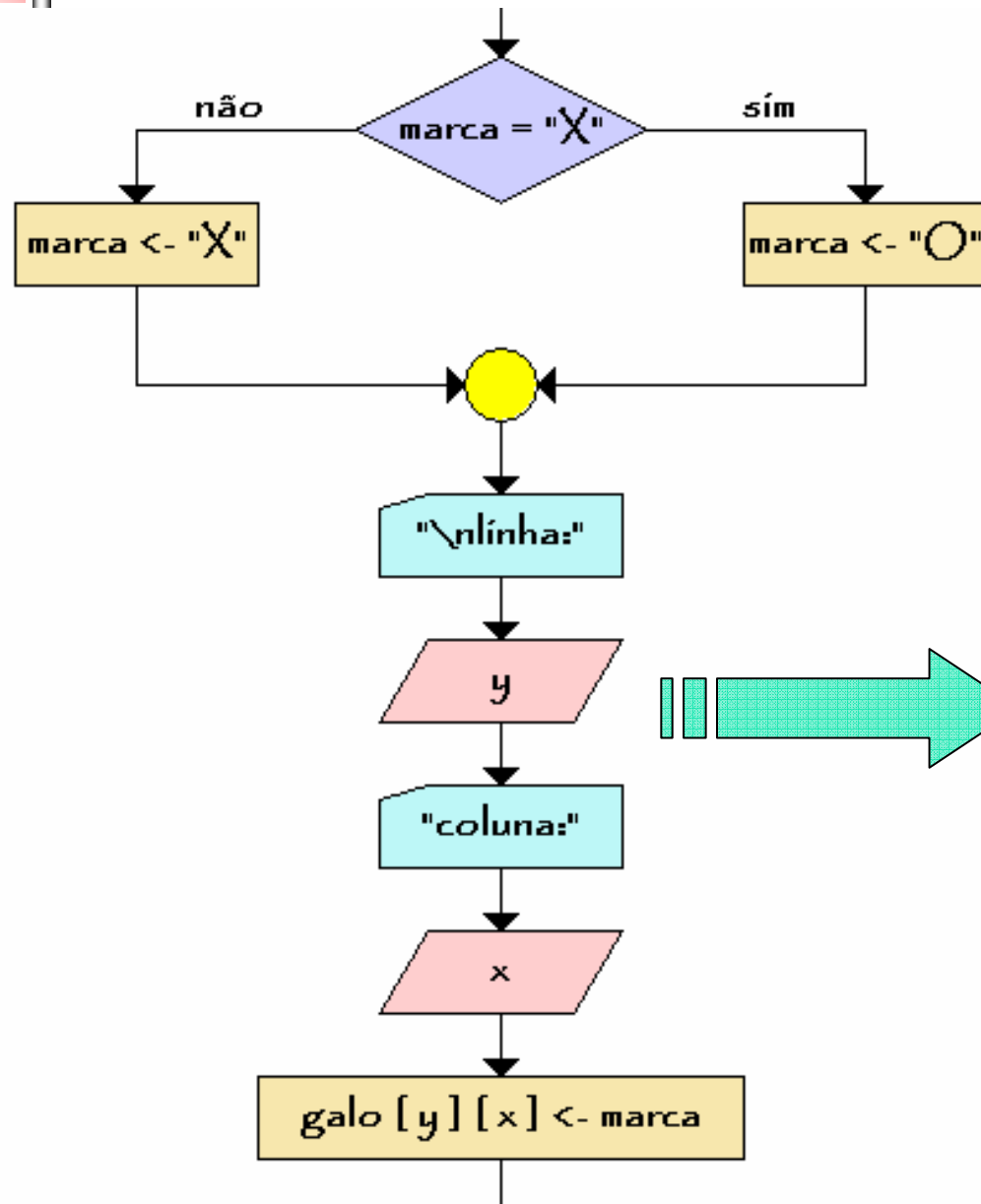




# Apresentar o Jogo



# Jogar





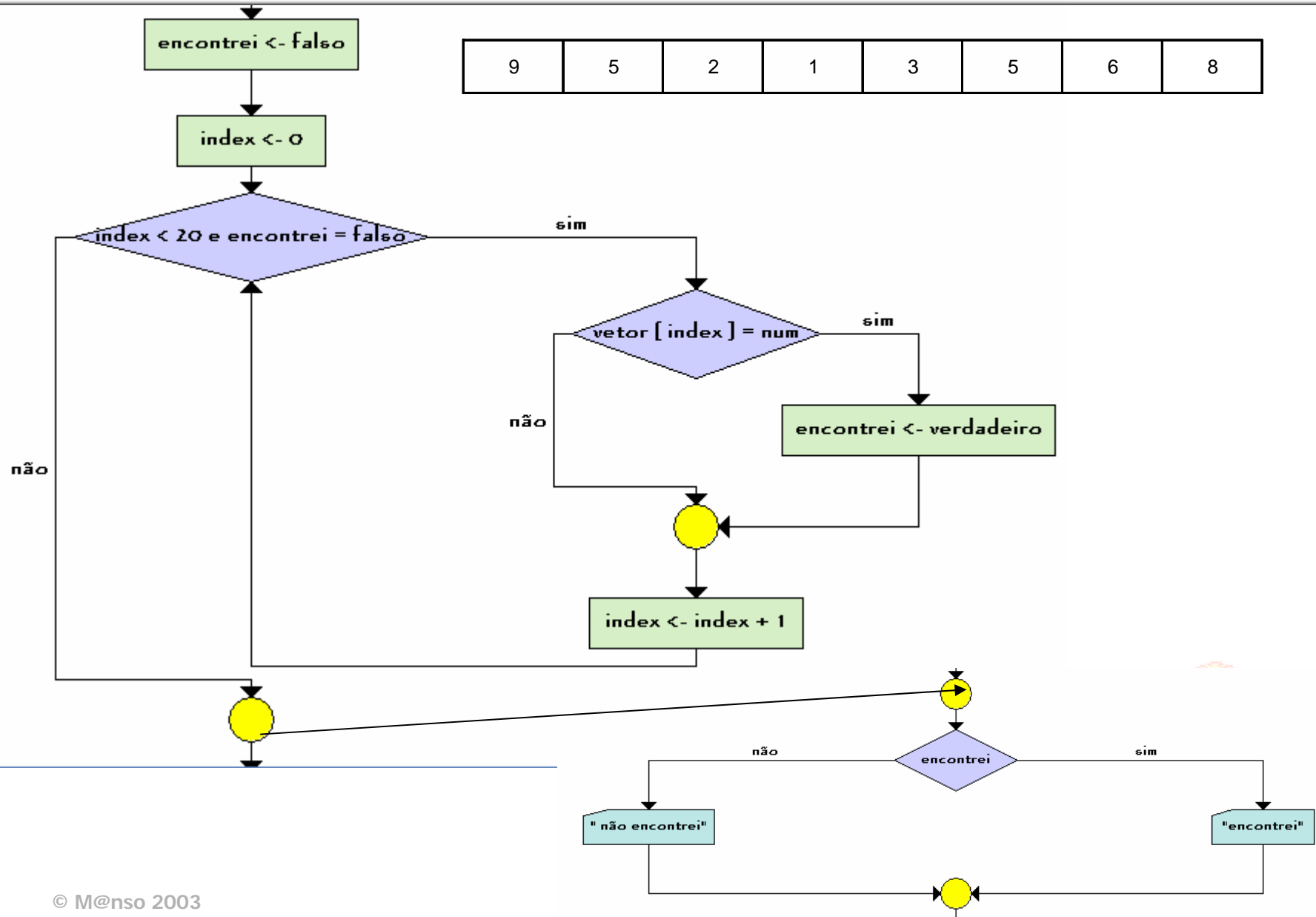
# Pesquisa e Ordenação

---



# Pesquisa sequencial

9	5	2	1	3	5	6	8
---	---	---	---	---	---	---	---



# Pesquisa sequencial

9	5	2	1	3	5	6	8
---	---	---	---	---	---	---	---

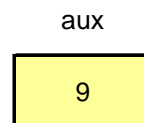
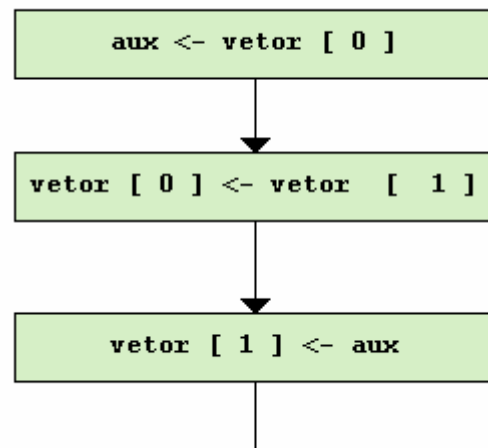
pesquisar

```
ler num
logico encontrei <- falso
index <- 0
enquanto index < 8 e encontrei = falso faz
  se vetor [ index ] = num entao
    encontrei <- verdadeiro
  fimse
  index <- index + 1
fimenquanto
se encontrei entao
  escrever "encontrei"
senao
  escrever " não encontrei"
fimse
```



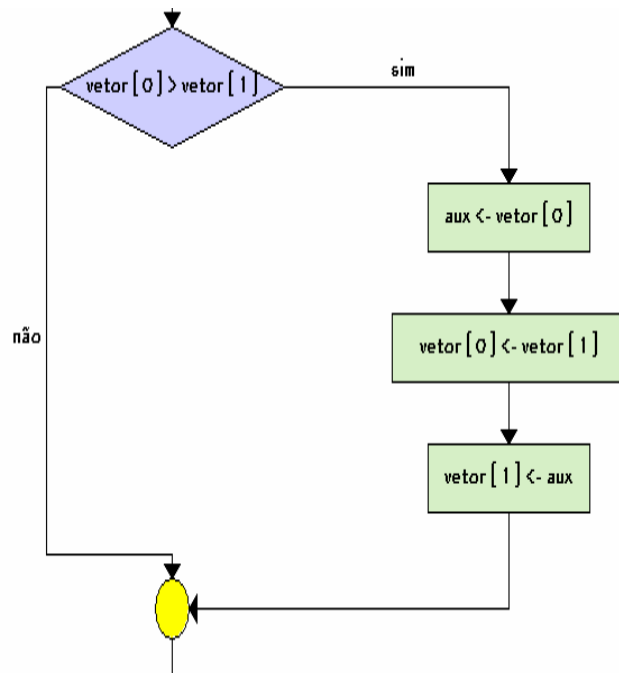
## Troca de valores dentro de arrays

- Construir um programa que troque os valores das duas primeiras posição de um vector

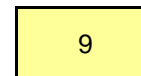


## Troca de valores dentro de arrays

- Construir um programa que ordene os valores das duas primeiras posição de um vector



aux

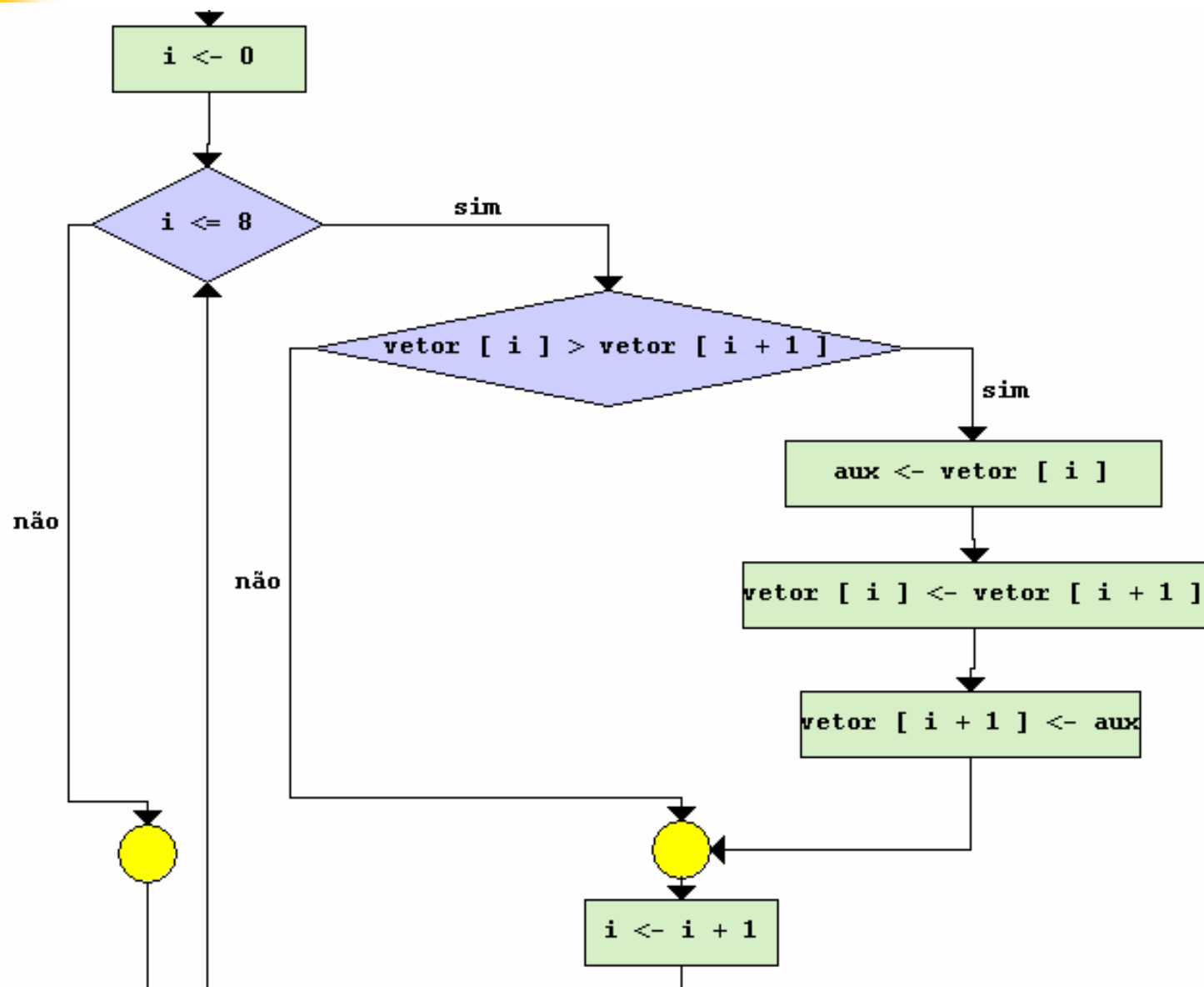


# Troca de valores dentro de arrays

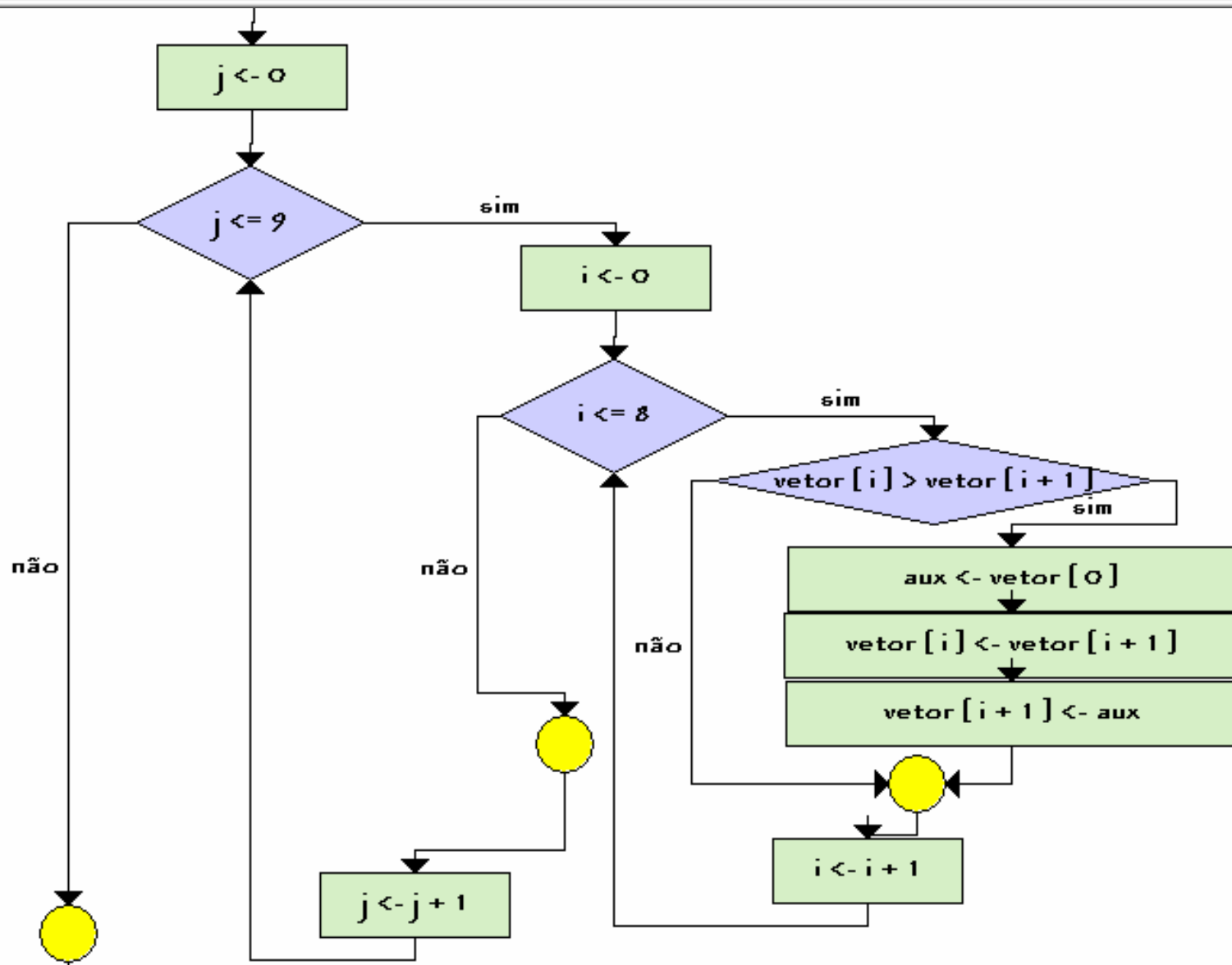
9	5	2	1	3	5	6	0
5	9	2	1	3	5	6	0
5	9	2	1	3	5	6	0
5	2	9	1	3	5	6	0
5	2	1	9	3	5	6	0
5	2	1	3	9	5	6	0
5	2	1	3	5	9	6	0
5	2	1	3	5	6	9	0
5	2	1	3	5	6	0	9



# Troca de valores dentro de arrays



# Bubble Sort



# Bubble Sort

## Bubble Sort

```
para j de 0 ate 9
  para i de 0 ate 8
    se vetor [ i ] > vetor [ i + 1 ] entao
      aux <- vetor [ i ]
      vetor [ i ] <- vetor [ i + 1 ]
      vetor [ i + 1 ] <- aux
    fimse
  proximo
proximo
```

## Resumo

- Arrays são conjunto de dados todos do mesmo tipo que são acessados através da indexação do mesmo nome
- É possível definir arrays de qualquer tipo de dados
- Os elementos dos arrays comportam-se como variáveis simples
- Os arrays têm uma ou mais dimensões
- Os indexes começam em **ZERO**
- O processamento de todos os elementos do Array envolve ciclos através da iteração dos índices
- A instrução LER le apenas uma variável
  - Os arrays representam muitas variáveis
  - Não é possível ler o array de uma só vez
- A instrução Escrever escreve uma variável ou expressão
  - Os arrays representam muitas variáveis
  - Não é possível escrever o array de uma só vez



# Dúvidas

