

No algoritmo o uso do var determina o tipo de parâmetro como por referência.

Construir um módulo que determine o maior e o menor número fornecidos dois números.

### *Passagem de parâmetro por ...*

```
Algoritmo "ex_maior_menor"  
  
Var  
    n1, n2, maior, menor : inteiro  
  
Procedimento maior_menor(a, b : inteiro)  
Inicio  
    se ( a > b )    entao  
        maior <- a  
        menor <- b  
    senao  
        maior <- b  
        menor <- a  
    fimse  
Fimprocedimento  
  
Inicio  
    leia(n1, n2)  
    maior_menor(n1, n2)  
    escreva(maior, menor)  
Fimalgoritmo
```

No módulo “maior\_menor” os parâmetros “a” e “b” são por valor.

## *Passagem de parâmetro por ...*

Algoritmo "ex\_maior\_menor"

Var

n1, n2, ma, me : inteiro

Procedimento maior\_menor(a, b : inteiro; var maior, menor : inteiro)

Inicio

se ( a > b )    entao

    maior <- a

    menor <- b

senao

    maior <- b

    menor <- a

fimse

Fimprocedimento

Inicio

    leia(n1, n2)

    maior\_menor(n1, n2, ma, me)

    escreva(ma, me)

Fimalgoritmo

**No módulo “maior\_menor” os parâmetros “a” e “b” são por valor e os parâmetros “maior” e “menor” são por referência.**

## *Mais exemplos*

Algoritmo Ex.

Var

a:vetor[1..100] de inteiro  
i, primeiro, ultimo : inteiro

```
procedimento prim_ult(v:vetor[1..100] de inteiro; var p, u : inteiro)  
inicio  
    p <- v[1]  
    u <- v[100]  
fimprocedimento
```

//rotina principal

inicio

para i de 1 ate 100 faca  
 leia( a[ i ] )  
fimpara  
prim\_ult( a, primeiro, ultimo)  
escreva(primeiro, ultimo)

finalalg

**No módulo “prim\_ult” o parâmetro “v” é por valor e os parâmetros “p” e “u” são por referência.**

Algoritmo Exx

Var

a:vetor[1..100] de inteiro  
i, quantos\_pares : inteiro

```
Procedimento p1(v:vetor[1..100] de inteiro; var n:inteiro)  
var  
    i : inteiro  
inicio  
    n <- 0  
    para i de 1 ate 100 faca  
        se v[i] mod 2 = 0 entao  
            n <- n + 1  
        fimse  
    fimpara  
fimprocedimento
```

//rotina

Inicio

para i de 1 ate 100 faca  
 leia( a[i] )  
fimpara  
p1(a,quantos\_pares)  
escreva(quantos\_pares)

FimAlg.

**No módulo “p1” o parâmetro “v” é por valor e o parâmetro “n” é por referência.**

Algoritmo Exxx

Var

a:vetor[1..100] de inteiro  
quantos\_pares : inteiro

```
Funcao f1(v:vetor[1..100] de inteiro):inteiro  
var  
    n, i : inteiro  
inicio  
    n <- 0  
    para i de 1 ate 100 faca  
        se v[i] mod 2 = 0 entao  
            n <- n + 1  
        fimse  
    fimpara  
    retorne(n)  
fimfuncao
```

//rotina

Inicio

para i de 1 ate 100 faca  
 leia( a[i] )  
fimpara  
quantos\_pares <- f1(a)  
escreva(quantos\_pares)

FimAlg.

**No módulo “f1” o parâmetro “v” é por valor.**

Algoritmo Exxxx

Var

v1, v2, v3:vetor[1..100] de inteiro  
i, quantos\_pares, quantos\_impares : inteiro

Procedimento p2(a:vetor[1..100] de inteiro; var b,c:vetor[1..100] de inteiro;  
var d, e : inteiro)

```
var  
  i : inteiro  
inicio  
  d <- 0  
  e <- 0  
  para i de 1 ate 100 faca  
    se a[i] mod 2 = 0 entao  
      d <- d + 1  
      b[d] <- a[i]  
    senao  
      e <- e + 1  
      c[e] <- a[i]  
    fimse  
  fimpara  
fimprocedimento
```

//rotina principal

Inicio

```
  para i de 1 ate 100 faca  
    leia( v1[i] )  
  fimpara  
  p2(v1, v2, v3, quantos_pares, quantos_impares)  
  para i de 1 ate quantos_pares faca  
    escreva( v2[i] )  
  fimpara  
  para i de 1 ate quantos_impares faca  
    escreva( v3[i] )  
  fimpara
```

FimAlgoritmo

**No módulo “p2” o parâmetro “a” é por valor e os parâmetros “b”, “c”, “d” e “e” são por referência.**