ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES 2per

MODULARIZAÇÃO + VETOR E MATRIZ

VETOR COMO PARÂMETRO DE SUB-ROTINAS

No algoritmo o vetor pode ser passado como parâmetro por valor (padrão) ou por referência (indicado por **VAR**).

Ex1.: uma função para calcular a média de n idades armazenadas em um vetor – parâmetros (de entrada) são o número de elementos do vetor e o próprio vetor com os valores.

```
FUNCAO idademedia (n: INTEIRO, idades: VETOR[1..10] DE
                    INTEIRO): INTEIRO
VAR
   i, soma: INTEIRO
INICIO
   soma <- 0
   PARA i DE 1 ATE n FAÇA
      soma <- soma + idades[i]
   FIMPARA
   idademedia <- soma DIV n
FIMFUNCAO
```

Ex2.: um procedimento para ler n nomes e armazená-los em um vetor – parâmetro de entrada é o número de elementos que devem ser lidos, e o parâmetro de saída é o próprio vetor preenchido com os valores.

```
PROCEDIMENTO lerVetNomes (n: INTEIRO, VAR nomes: VETOR [1..10
                          DE CARACTERE])
VAR
   i: INTEIRO
INICIO
   PARA i DE 1 ATE n FAÇA
     ESCREVA("Nome: ")
     LEIA(nomes[i])
   FIMPARA
FIMPROCEDIMENTO
```

Em C++ um vetor é **sempre passado por referência**!! Ainda, não há necessidade em indicar o tamanho máximo do vetor ([ ] fica vazio).

```
int idadeMedia (int n, unsigned int idades[10]){
    int soma=0;
    for(int i=0; i<n; i++)
        soma = soma + idades[i];
    return soma/n;
}

void lerVetNomes (int n, string nomes[]){
    for(int i=0; i<n; i++)    {
        cout << "Nome: ";
        getline(cin,nomes[i]);
    }
}
```

MODULARIZAÇÃO + VETOR E MATRIZ

VETOR COMO PARÂMETRO DE SUB-ROTINAS

EXEMPLO:

```
ALGORITMO "EXEMPLOSUBPROGRAMAS"
FUNCAO leiaNumInteiroIntervalo (linf, lsup:INTEIRO): INTEIRO
VAR
   n: INTEIRO
INICIO
   REPITA
      ESCREVA("Qtde de elementos(",linf,"-",lsup,"): ")
      LEIA(n)
   ATE (n >= linf E n <= lsup)
   leiaNumInteiroIntervalo <- n
FIMFUNCAO

PROCEDIMENTO leiaVetInteiro (n: INTEIRO, VAR numeros: VETOR[1..15]
                             DE INTEIRO)
VARIAVEIS INTEIRO i
INICIO
   PARA i DE 1 ATE n FAÇA
      ESCREVA(i, "o. elemento: ")
      LEIA(numeros[i])
   FIMPARA
FIMPROCEDIMENTO

FUNCAO achaMenor (n: INTEIRO, numeros: VETOR[1..15] DE INTEIRO):
                  INTEIRO
VAR
   i, menor: INTEIRO
INICIO
   menor <- numeros[1]
   PARA i DE 2 ATE n FAÇA
      SE menor > numeros[i] ENTAO
         menor <- numeros[i]
      FIMSE
   FIMPARA
   achaMenor <- menor
FIMFUNCAO

VAR
   n: INTEIRO
   v: VETOR[1..15] DE INTEIRO
INICIO
   n <- leiaNumInteiroIntervalo(1,15)
   leiaVetInteiro(n, v)
   ESCREVA("Menor elemento ", achaMenor(n,v))
FIMALGORITMO
```

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES 2per

MODULARIZAÇÃO + VETOR E MATRIZ

VETOR COMO PARÂMETRO DE SUB-ROTINAS

EXEMPLO:

```
#include <iostream>
using namespace std;

int leiaNumInteiroIntervalo (int linf, int lsup){
    int n;
    do{
        cout<<"Qtde de elementos("<<linf<<"-"<<lsup<<"): ";
        cin>>n;
    }while (not(n >= linf and n <= lsup));
    return n;
}

void leiaVetInteiro (int n, int numeros[]){
    for(int i=0; i<n; i++)    {
        cout<<i+1<<"o. elemento: ";
        cin>>numeros[i];
    }
}

int achaMenor (int n, int numeros[]){
    int menor;
    menor = numeros[0];
    for(int i=1; i<n; i++)
        if(menor > numeros[i])
            menor = numeros[i];
    return menor;
}

int main(){
    int n, v[15];
    n = leiaNumInteiroIntervalo(1,15);
    leiaVetInteiro(n,v);
    cout<<"Menor elemento "<<achaMenor(n,v)<<endl;
    return 0;
}
```

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES 2per

MODULARIZAÇÃO + VETOR E MATRIZ

MATRIZ COMO PARÂMETRO DE SUB-ROTINAS

No algoritmo, assim como o vetor, a matriz pode ser passada como parâmetro por valor (padrão) ou por referência (indicado por **VAR**).

Ex3.: um procedimento para preencher uma matriz n x m com valores reais – parâmetros de entrada são as dimensões da matriz e o parâmetro de saída é a própria matriz preenchida com os valores.

```
PROCEDIMENTO lerMatriz (n, m: INTEIRO, VAR matA: VETOR
                        [1..10,1..10] DE REAL])
VAR
   i,j: INTEIRO
INICIO
   PARA i DE 1 ATE n FAÇA
     PARA j DE 1 ATE m FAÇA
        ESCREVA("Elemento[",i,",",j,"]: ")
        LEIA(matA[i,j])
     FIMPARA
   FIMPARA
FIMPROCEDIMENTO
```

Em C++ uma matriz é **sempre passado por referência também**!!
Porém, é preciso indicar pelo menos o número máximo de colunas que a matriz tem ([ ][valor]), podendo deixar em branco apenas o número máximo de linhas.

```
void lerMatriz (int n, int m, float matA[][10]){
    for(int i=0; i<n; i++)
        for(int j=0; j<m; j++){
            cout<<"Elemento["<<i+1<<","<<j+1<<"]: ";
            cin>>matA[i][j];
        }
}
```

Fixação do conteúdo: ler o cap9 Subprogramas do livro Algoritmos e Programação com Exemplos em Pascal e C [Série Livros didáticos informática UFRGS], de Edelweiss, Nina.
Dar atenção a seção 9.7 exercícios de fixação (exemplos resolvidos).

Atividade prática: refazer os exercícios das listas de vetor e matriz de Algoritmos 1per, reorganizando o código em subprogramas.