PROGRAMAÇÃO

FUNÇÕES NA LINGUAGEM C

Funções em C

Implementações básicas :

a) Implementar a função ***LerValorValido*** que verifica se um valor introduzido pelo utilizador pertence ao conjunto limitado por dois dados valores inteiros, devolvendo o primeiro valor que pertença àquele intervalo;

b) Implementar o procedimento ***LerVector*** que preenche um vector com **Dim** valores reais (Dim ≥ 1);

a) Implementar o procedimento ***MostrarVector*** que mostra um vector com **Dim** valores reais (Dim ≥ 1);

Funções em C

Função: *LerValorValido*

Parâmetros:

- Dois valores inteiros, **inf** e **sup**, correspondentes aos limites inferior e superior do intervalo válido

Retorno:

- Um número inteiro pertencente ao conjunto **{inf, ..., sup}**

Funções em C

Algoritmo:

Fazer

Mostrar a mensagem "**Inserir um inteiro entre inf e sup**"

Introduzir um valor **inteiro** para **Dim**

Enquanto 'Dim < inf ou Dim > sup' é verdadeira

Devolver (Dim)

Funções em C

```
int LerValorValido (int inf, int sup)
{
   int   Dim;
   do {
      printf ("Insira um inteiro entre %d e %d: ", inf, sup);
      scanf ("%d", &Dim);
   } while (Dim < inf || Dim > sup);
   return (Dim);
}
```

Funções em C

Procedimento: *LerVector*

Parâmetros:

- Um vector **X** vazio (sem valores)
- Um número inteiro **Dim** (≥ 1), que é o tamanho de **X**

Saída:

- O vector **X** preenchido (com valores)

Funções em C

Algoritmo:

Para **k** com valores desde **0** até **Dim-1** Fazer

Mostrar a mensagem "**Inserir um valor real:** "

Introduzir um valor **real** para **X[k]**

Funções em C

```
void LerVector (float X[], int Dim)
{
    int k;
    for (k = 0; k <= Dim-1; k++)
    {
        printf ("Inserir um valor real: ");
        scanf ("%f", &X[k]);
    }
}
```

Funções em C

Procedimento: *MostrarVector*

Parâmetros:

- O vector **X** preenchido (com valores)
- Um número inteiro **Dim** (≥ 1), que é o tamanho de **X**

Saída:

- Apresenta no écran os valores do vector **X**

Funções em C

Algoritmo:

Para **k** com valores desde **0** até **Dim-1** Fazer

Mostrar o valor de **X[k]**

Funções em C

```
void  MostrarVector  (float  X[],  int  Dim)
{
    int  k;
    for (k = 0; k <= Dim-1; k++)
        printf ("%7.2f ", X[k]);      // com 2 casas décimais
}
```

Funções em C

Enunciado:

a) Implementar a função ***PosicaoMaior*** que determina (e devolve) a posição/índice do maior valor de um vector **X** com **Dim** valores reais;

b) Construir um programa em C que leia um vector **V** com **N** valores reais (N ≥ 1), determine a posição/índice do maior dos seus valores, calcule o maior elemento e escreva no écran os resultados obtidos, usando as funções construídas anteriormente.

Funções em C

Função: *PosicaoMaior*

Parâmetros:

- Um vector **X** preenchido com valores reais
- Um número inteiro **Dim** (≥ 1), que é o tamanho de **X**

Retorno:

- Um valor inteiro correspondente à posição/índice do maior valor do vector **X**

Funções em C

Algoritmo:

Atribuir a **pos_maior** o valor **inteiro 0**

Para **k** com valores desde **1** até **Dim-1** Fazer

Se 'X[k] > X[pos_maior]' é verdadeira Então

Atribuir a **pos_maior** o valor **inteiro k**

Devolver (pos_maior)

Funções em C

```
int  PosicaoMaior  (float  X[],  int  Dim)
{
    int  k, pos_maior;
    pos_maior = 0;  // assumir que o maior está na 1ª posição
    for (k = 1; k <= Dim-1; k++)
        if (X[k] > X[pos_maior])
            pos_maior = k;
    return (pos_maior);
}
```

Funções em C

Algoritmo (principal):

Atribuir a **N** o valor **inteiro LerValorValido(1, 50)**

LerVector (V, N)

Atribuir a **pos_maior** o valor **inteiro PosicaoMaior (V, N)**

Mostrar a mensagem "**O maior valor está na posição :** "

Mostrar o valor de **pos_maior**

Atribuir a **maior** o valor **real V[pos_maior]**

Mostrar a mensagem "**O maior valor é :** "

Mostrar o valor de **maior**

Funções em C

```
#include <stdio.h>
int  LerValorValido (int  inf,  int  sup);
void LerVector (float X[], int Dim);
int  PosicaoMaior  (float  X[],  int  Dim);
main ( )
{
    float   V[50], maior;
    int     N, pos_maior;
    N = LerValorValido(1, 50);
    LerVector (V, N);
    pos_maior = PosicaoMaior (V, N);
    printf ("O maior valor está na posição : %d.\n", pos_maior);
    maior = V[pos_maior];
    printf ("O maior valor é : %f.\n", maior);
}
```

Funções em C

Enunciado:

a) Implementar um procedimento para inverter um vector **X** com **Dim** valores reais (trocar o 1º com o último, o 2º com o penúltimo, ...);

b) Construir um programa em C que leia um vector **V** com **N** valores reais (N ≥ 1), inverta este vector e escreva no écran o novo vector, usando as funções implementadas anteriormente.

Funções em C

Procedimento: *InverterVector*

Parâmetros:

- Um vector **X** preenchido com valores reais
- Um número inteiro **Dim** (≥ 1), que é o tamanho de **X**

Saída:

- O vector **X** invertido

Funções em C

Algoritmo:

Para **k** com valores desde **0** até **(Dim/2)-1** Fazer

Atribuir a **aux** o valor **real X[k]**

Atribuir a **X[k]** o valor **real X[(Dim-1)-k]**

Atribuir a **X[(Dim-1)-k]** o valor **real aux**

Funções em C

```
void InverterVector (float X[], int Dim)
{
    int k;
    float aux;
    for (k = 0; k <= (Dim/2)-1; k++)
    {
        aux = X[k];
        X[k] = X[Dim-1-k];
        X[Dim-1-k] = aux;
     }
}
```

Funções em C

Algoritmo (principal):

Atribuir a **N** o valor **inteiro LerValorValido(1, 50)**

LerVector (V, N)

InverterVector (V, N)

Mostrar a mensagem **"Vector invertido"**

MostrarVector (V, N)

Funções em C

```
#include <stdio.h>
int  LerValorValido (int  inf, int  sup);
void LerVector (float  X[], int  Dim);
void MostrarVector (float  X[], int  Dim);
void InverterVector (float  X[],  int  Dim);
main ( )
{
    float   V[50];
    int  N;
    N = LerValorValido(1, 50);
    LerVector (V, N);
    InverterVector (V, N);
    printf("Vector invertido:\n");
    MostrarVector (V, N);
}
```