

Declarando vetores

A declaração

```
int a[10];
```

define um vetor ("array") de tamanho 10, isto é, um bloco de 10 objetos consecutivos que recebem os nomes a[0], a[1], a[2], a[9].

NOTE que, em C, o primeiro índice é 0 (zero) (dependendo da linguagem, o primeiro índice pode ser 1).

Arrays podem ser de quaisquer tipos.

double x[1000] define um vetor de tamanho 1000 para guardar 1000 números no formato double: x[0], x[1], x[2], ... x[999].

EXEMPLO 1. Um programinha que lê dez números, e imprime o menor e o maior:

```
#include <stdlib.h>
#include <stdio.h>

int main ()
{
    double x[10], maior, menor;
    int i;

    for (i=0;i<=9;i++) scanf ("%lf", &x[i]);
    maior = menor = x[0];
    for (i=0;i<=9;i++){ if (x[i] < menor) menor = x[i];
                       if (x[i] > maior) maior = x[i]; }
    printf("\nO menor numero eh %g e o maior eh %g\n", menor, maior);
    system("pause");
    return(0);
}
```

EXEMPLO 2. O mesmo programa, lendo números até que o usuário sinalize o final (Ctrl-z):

```
#include <stdlib.h>
#include <stdio.h>

int main ()
{
    double x[100], maior, menor;
    int i, N;

    i=0;
    while (scanf("%lf", &x[i]) != EOF){N=i; i++;}
    maior = menor = x[0];
    for (i=0;i<=N;i++){ if (x[i] < menor) menor = x[i];
                       if (x[i] > maior) maior = x[i]; }
    N++;
    printf("\nVoce entrou com %d numeros; o menor eh %g e o maior eh %g\n", N, menor, maior);
    system("pause");
    return(0);
}
```

EXERCÍCIO 1. Modifique o programa acima para evitar que o usuário entre com mais de 100 números.

EXERCÍCIO 2. Faça com que o programa conte quantos números são negativos e quantos são positivos.

EXERCÍCIO 3. Faça com que o programa normalize os números com respeito ao módulo do maior, isto é, substitua cada número x_i por $x_i/|X|$, onde $|X|$ é o maior valor absoluto de todos os números. (TESTE se $|X|$ não é zero!!!)

EXERCÍCIO 4. Faça com que o programa normalize os números com respeito à norma euclidiana deles, isto é, cada um deles deve ser multiplicado por um fator α , positivo, de modo que a soma dos quadrados de todos seja exatamente 1 $\left(\sum_{i=1}^N x_i^2 = 1\right)$. (NOTE que deve haver pelo menos um número diferente de zero – teste isso!)

EXERCÍCIO 5. Insira no programa do Exercício 1 o cálculo do valor médio dos números, $\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$.

EXERCÍCIO 6. Faça com que o programa do Exercício 5 conte quantos números da lista são menores que a média \bar{x} . Imprima a porcentagem de números que são menores que a média.

EXERCÍCIO 7. Insira nesse mesmo programa o cálculo do desvio padrão σ , sabendo que $\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2$. Faça com que o programa conte quantos números da lista estão no intervalo $[\bar{x} - \sigma, \bar{x} + \sigma]$. Imprima o resultado em porcentagem.

EXERCÍCIO 8. Calcule o desvio padrão, no programa, pela fórmula alternativa $\sigma^2 = \frac{N}{N-1} (\overline{x^2} - \bar{x}^2)$.

EXEMPLO 3. Um algoritmo de ordenação: ordenando os números do menor para o maior:

```
#include <stdlib.h>
#include <stdio.h>

int main ()
{
    double x[10], temp;
    int i, N, trocou;

    i=0;
    while (scanf("%lf", &x[i]) != EOF){N=i; i++;}
    trocou = 1;
    while (trocou) {
        trocou = 0;
        for (i=0; i<N; i++)
            if (x[i]>x[i+1]) { temp = x[i];
                            x[i] = x[i+1];
                            x[i+1] = temp;
                            trocou = 1; }
    }
    printf("\n Os numeros ordenados sao:\n");
    for (i=0; i<N; i++)printf(" %g", x[i]);
    printf("\n");

    system("pause");
    return(0);
}
```

EXERCÍCIO 9. Modifique o programa acima para ordenar os números em ordem decrescente.

EXERCÍCIO 10. Faça um programa que, após ler uma quantidade de números,

- (1) Ordene os mesmos em ordem crescente;
- (2) Leia um número extra e insira esse número entre os antigos, mantendo a ordenação.