

VETORES

Motivação:

Até o momento utilizamos variáveis que armazenam apenas um valor de cada vez, mas existem situações em que precisamos armazenar diversos valores como por exemplo:

A relação de aluno e nota em ordem decrescente de nota;

Uma “variável” que armazena o nome “PEDRO” também é um vetor do tipo char.

Considere um programa que leia a nota de 3 alunos e imprima a média:

```
#include<stdio.h>
int main()
{
    float nota1, nota2, nota3, media;

    printf("Digite a primeira nota: ");
    scanf("%f", &nota1);
    printf("Digite a segunda nota1: ");
    scanf("%f", &nota2);
    printf("Digite a terceira nota: ");
    scanf("%f", &nota3);

    media = (nota1 + nota2 + nota3 ) / 3;
    printf("Notas: %f %f %f %f\n", nota1, nota2, nota3);
    printf("Media: %f\n", media);
    return 0;
}
```

Como ficaria o programa se em vez de 3 fossem 200 notas?

Vetor é uma variável homogênea unidimensional que armazena muitas informações do mesmo tipo primitivo.

Cria um vetor de 10 posições (0 até 9) do tipo float.

```
float notas[10];
```

0	1	2	3	4	5	6	7	8	9

O exemplo abaixo preenche o vetor com 10 notas. Observe que para fazer referência a uma posição no vetor utilizamos um índice que começa em 0 (zero).

```
for(i=0;i<10;i++)  
{  
    printf("\nDigite uma nota: ");  
    scanf("%f",&notas[i]); // digite: 8, 7.5, 4, 6.5, 2, 9.5, 5.7, 7, 4, 6  
}
```

8.0	7.5	4.0	6.5	2.0	9.5	5.7	7.0	4.0	6.0
0	1	2	3	4	5	6	7	8	9

Cria um vetor com 6 posições para armazenar a quantidade de aspirantes por quarto.

```
int quartos[] = {39,39,38,38,38,37};
```

39	39	38	38	38	37
0	1	2	3	4	5

Exercício: Escreva um programa que leia 10 notas, imprima a média e em seguida a relação de notas.

```
#include<stdio.h>
int main()
{
    int i;
    float somanota = 0, float notas[10];

    for(i=0;i<10;i++)
    {
        printf("\nDigite uma nota: ");
        scanf("%f",&notas[i]);
    }

    for(i=0;i<10;i++)
        somanota = somanota + notas[i];

    printf("\nMedia: %4.1f", somanota/10);

    for(i=0;i<10;i++)
    {
        printf("\n %4.1f",notas[i]);
    }
}
```

Lista de Exercícios

- 18) Construir um algoritmo para ler 120 números, armazená-los num vetor, inverter a posição de seus elementos e imprimir o resultado.
- 19) Escreva um programa que preencha a partir do teclado 2 vetores $r[10]$ e $s[20]$, escreva um algoritmo que gere o vetor “x”, união de r e s .
- 20) Escreva um programa que inicie 2 vetores $r[10]$ e $s[20]$ com valores, escreva um algoritmo que gere o vetor “z” com os elementos de r que não estão em s .
- 21) Dados 2 vetores $r[10]$ e $s[20]$ preenchidos com valores, escreva um algoritmo que gere o vetor “w”, da seguinte forma:
- $w[0]=r[0]; w[1]=s[0]; w[2]=r[1]; w[2]=s[1]; \dots w[19]=r[9];$
 $w[19]=s[9]; w[20]=s[10]; w[21]=s[11]; \dots w[29]=s[19];$
- 22) Fornecidos pelo teclado números inteiros e positivos contidos no intervalo de 1 até 10, em qualquer ordem, sendo o último um flag = 99, pede-se: imprimir para cada número da série, a quantidade de vezes que ele foi digitado. caso algum número da série não seja digitado, imprimir a mensagem: “o número x não foi digitado nenhuma vez”.

23) Em uma eleição a partir de uma relação prévia de candidatos numerados de 1 a 30, cada aspirante deverá digitar o seu voto pelo número do candidato. Para encerrar a votação será utilizado um flag = 99. Discutir o caso de empate.

24) Fornecidos pelo teclado um primeiro conjunto de 100 números inteiros e positivos, armazená-los em um vetor. fornecidos, a seguir, mais números inteiros e positivos, indicar para cada um destes novos números sua posição dentro do vetor. se o número não for encontrado imprima a mensagem: “o número x não está no vetor”. Um flag = 0 encerra a entrada do segundo conjunto de números.