#### Vetores e Matrizes em C

***Soluções***

**Introdução**

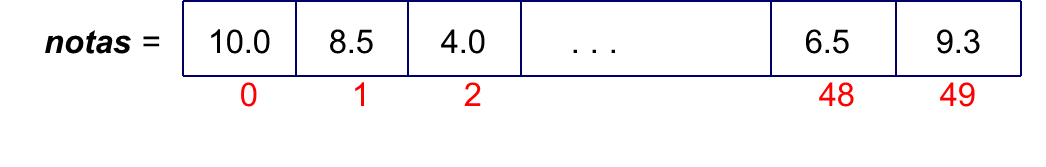
Nos capítulos anteriores focou-se muito nas estruturas de controle de execução de programas na linguagem C. Foram estudadas e praticadas estruturas sequenciais, estruturas de decisão e seleção e de repetição. Entretanto a manipulação da memória se restringiu ao uso de variáveis de tipos primitivos, que permitem o armazenamento de um único valor a cada instante de tempo.

Este capítulo vai focar uma forma de armazenamento de coleções de dados. Neste contexto, uma importante estrutura de armazenamento de informações na memória principal são os vetores e matrizes. Estes são conhecidos como ***estruturas homogêneas***, uma vez que permitem armazenamento e manipulação de dados de um mesmo tipo.

Como exemplo, podemos citar a necessidade de tratar listas de dados, como:

* + - Notas de uma turma de 50 alunos;
    - Nomes de 1.000 produtos vendidos em uma loja;
    - Cadeia de 100 caracteres.

Todos os casos acima são adequadamente representados por um ***arranjo*** ou ***vetor***. Graficamente pode-se representar um vetor como na figura a seguir, que descreve o exemplo das notas da turma de alunos:



Diz-se então que *notas* é uma estrutura ***homogênea unidimensional*** de dados:

* + - * ***Homogênea***: pois todos os componentes são do mesmo **tipo de dados** (*float* ou *double*);
      * ***Unidimensional***: pois os elementos indexados em uma única linha, formando uma sequência de dados.
      * Na figura, as caixinhas representam cada elemento de dados e a numeração abaixo das caixinhas representam o índice de cada elemento, que, na linguagem C começa sempre pela posição **0**. Por exemplo, o elemento de índice **1** armazena o valor de **nota = 8.5**.
      * O vetor utiliza sempre posições contíguas de memória.

**Vetores na Linguagem C**

A declaração de um vetor na linguagem C é muito parecida com a declaração de uma variável simples ou não estruturada, bastando apenas acrescentar entre colchetes o tamanho do vetor, ou seja, a quantidade de elementos que o programandor deseja que o vetor tenha.

**Sintaxe**:

**tipo nome[tamanho];**

**Semântica:**

É alocada na memória, em posições contíguas, uma variável de nome ***nome*** e área igual à quantidade de bytes do tipo ***tipo*** vezes o ***tamanho***declarado*.*

**Exemplo:**

**float notas[50];**

**int nums [10];**

**char palavra [20];** *// neste caso, um exemplo de vetor que forma uma “string”*

**Referenciando elementos do vetor na lógica de programas C:**

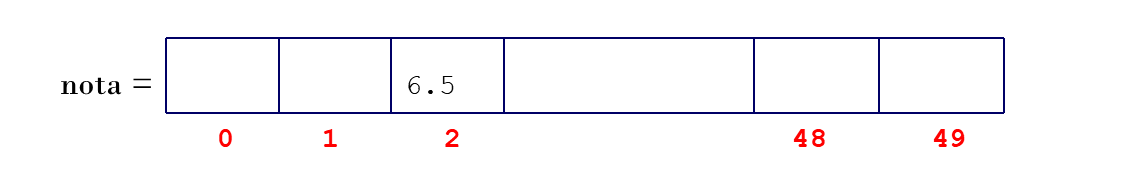
Um vetor, como qualquer variável, é declarado para armazenar dados e ser manipulado pelos programas. Porém, o vetor não é manipulado como um todo. O que é manipulável é cada elemento do vetor, um de cada vez. Para referenciar o elemento do vetor, basta escrever o nome do vetor, seguido da posição do elemento, entre colchetes.

**Exemplo**:

Atribuido valor **6.5** ao elemento de índice **2** do vetor *nota*:

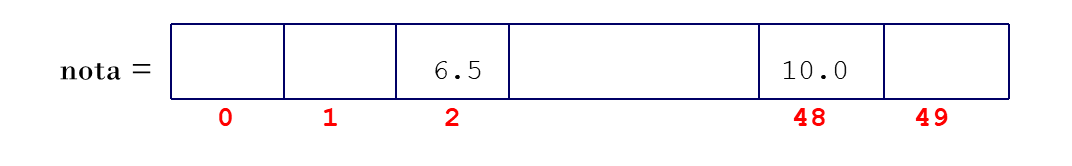
nota[2] = 6.5;

Esta instrução produz o seguinte resultado:



Agora, armazenando na posição **48** do vetor *nota* um valor informado pelo teclado:

scanf(“%d”, &nota[48]);// suponha que seja informado o valor **10.0**



Elementos de vetores também podem ser utilizados em expressões aritméticas ou lógicas, como qualquer variável. Exemplo: usando um elemento do vetor *nota* como operando de uma expressão:

for(int i = 0; i < 50; i++)

Soma = Soma + **nota[i]**;

Exemplo de um programa completo que lê a nota de 20 alunos e depois verifica os alunos aprovados, isto é, alunos com nota ≥ 5,0:

void main (){

float nota[20];

int i;

for(i = 0; i < 20; i++){ // laço que lê as 20 notas e carrega o vetor

printf(“\nInforme nota do aluno nr. %d: “, i+1);

scanf(“%d”, &nota[i]);

}

for(i = 0; i < 20; i++){ // laço que verifica os aprovados/reprovados

if(nota[i] >= 5)

printf(“\nAluno nr. %d aprovado com %.1f.“, i+1,nota[i]);

else

printf(“\nAluno nr. %d reprovado com %.1f.“,i+1,nota[i]);

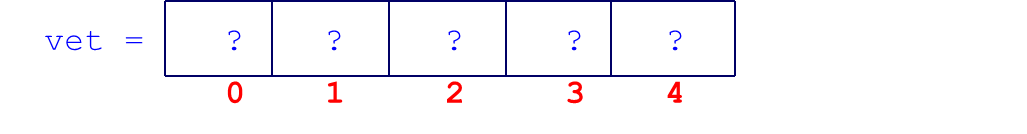
}

}

**Inicialização de vetor na linguagem C:**

Ao ser declarado, um vetor não é iniciado, ficando seus elementos preenchidos com valores indefinidos. Exemplo:

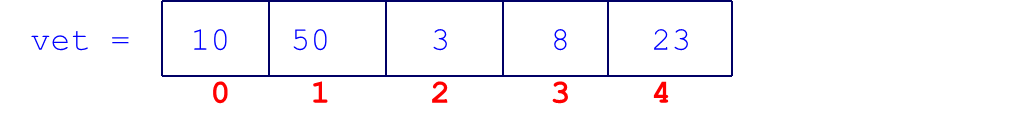
int vet[5];



Para iniciar um vetor, pode-se utilizar a construção a seguir:

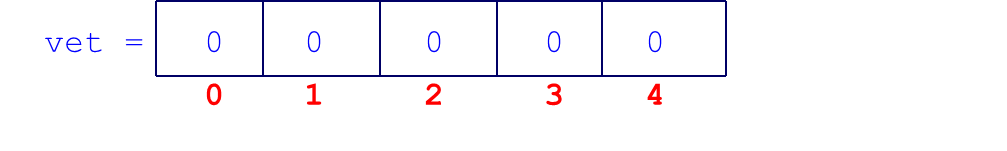
int vet[5] = {10, 50, 3, 8, 23};

Neste caso, o vetor é devidamente iniciado como na imagem abaixo:

****

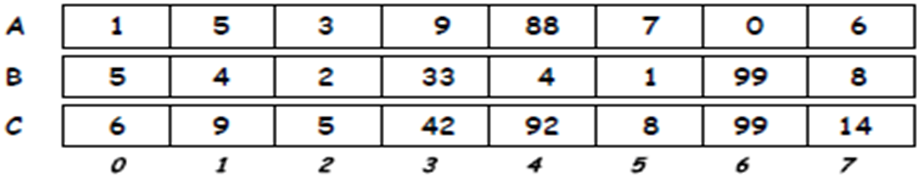
Se definir o vetor nas classes de armazenamento ***static*** ou ***extern***, o vetor é iniciado com valor *default* do tipo. Exemplo:

static int vet[5];

****

**Exemplo de um programa completo aplicando *vetor*:**

Programa que lê dados de dois vetores, A e B, com oito elementos inteiros cada e gera um 3º vetor, C, com a soma dos elementos dos dois, mostrando seu conteudo. Exemplo:



void main(){

static int A[] = {1, 5, 3, 9,88, 7, 0, 6};

static int B[8] = {5, 4, 2, 3, 4, 1,99, 8};

int i, C[8];

for(i = 0; i < 8; i++)

C[i] = A[i] + B[i];

printf(“\n C = { “);

for(i = 0; i < 8; i++

printf(“%d, “, C[i]);

printf(“ }; \n“);

}

**Exercícios aplicando vetores**

1. Faça um programa em C que leia pelo teclado as médias finais de uma turma com 15 alunos e mostre a quantidade de alunos com conceito A (médias >= 9), quantidade de alunos aprovados (médias >= 5), quantidade de alunos reprovados (médias < 5) e quantidade de alunos sem rendimento (média = 0). Ainda calcule e mostre a média da turma e as notas de todos os alunos acima da média.

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

void main(){

float notas[15], media, soma = 0;

int i, concA = 0; int aprov = 0; int reprov = 0; int sr = 0;

*// GERAR NOTAS DE FORMA ALEATÓRIA*

srand(time(NULL));

for(i = 0; i < 15; i++){

notas[i] = (rand()%100)/10.0;

printf("\n Nota %2d: %.1f", i+1, notas[i]);

}

*// CONTABILIZAR NOTAS*

for(i = 0; i < 15; i++){

soma += notas[i];

if(notas[i] >= 5){

aprov++;

if(notas[i] >= 9)

concA++;

} else {

reprov++;

if(notas[i] == 0)

sr++;

}

}

media = soma/15.0;

printf("\n\n Media da turma: %.1f.", media);

printf("\n Alunos com nota acima da media: ");

for(i = 0; i < 15; i++)

if(notas[i] > media)

printf("\n Nota %2d: %.1f", i+1, notas[i]);

printf("\n\n Quantidade alunos com conceito A: %d.", concA);

printf("\n Quantidade alunos aprovados: %d.", aprov);

printf("\n Quantidade alunos reprovados: %d.", reprov);

printf("\n Quantidade alunos sem rendimento: %d.", sr);

}

1. Faça um programa C que leia um conjunto X com 20 elementos reais e calcule a diferença entre o maior e o menor elemento existente, bem como as posições que os mesmos ocupam no conjunto. Esta diferença entre o maior e menor elementos de um conjunto de valores é denominada *amplitude*.

void main(){

float x[20];

int i, posMaior = 0; int posMenor = 0;

*// GERAR VALORES ALEATÓRIOS PARA x[i]*

srand(time(NULL));

for(i = 0; i < 20; i++){

x[i] = (rand()%1000)/10.0;

printf("\n x[%02d] = %4.1f.", i, x[i]);

}

*// ENCONTRAR POSIÇÃO DO MAIOR E DO MENOR*

for(i = 1; i < 20; i++){

if(x[i] > x[posMaior])

posMaior = i;

else

if(x[i] < x[posMenor])

posMenor = i;

}

printf("\n\n Maior valor: x[%02d] = %4.1f.", posMaior, x[posMaior]);

printf("\n Menor valor: x[%02d] = %4.1f.", posMenor, x[posMenor]);

printf("\n Amplitude: %4.1f.\n\n", x[posMaior]-x[posMenor]);

}

1. Faça um programa C que tenha como entrada um vetor com 10 elementos inteiros e substitua todos os elementos negativos por 0 e depois mostre na tela o conteúdo do vetor corrigido.

void main(){

int vet[10];

int i;

*// GERAR VALORES ALEATÓRIOS PARA vet[i]*

srand(time(NULL));

for(i = 0; i < 10; i++){

vet[i] = (rand()%200) - 100;

printf("\n vet[%d] = %3d.", i, vet[i]);

}

*// ELIMINAR VALORES NEGATIVOS*

printf("\n");

for(i = 0; i < 10; i++){

if(vet[i] < 0 )

vet[i] = 0;

printf("\n vet[%d] = %2d.", i, vet[i]);

}

printf("\n\n");

}

1. Faça um programa C que leia todos os elementos de um vetor V com MAX elementos inteiros e um valor K qualquer. Após, determine e imprima o vetor resultante da multiplicação de V por K.

void main(){

const int MAX = 10;

int V[MAX], i, K;

printf("Informe o valor de K: ");

scanf("%d", &K);

*// GERAR VALORES ALEATÓRIOS PARA vetor[i]*

srand(time(NULL));

for(i = 0; i < MAX; i++){

V[i] = (rand()%20);

printf("\n vetor[%d] = %2d.", i, V[i]);

}

*// CALCULAR E MOSTRAR (V[i] x K)*

printf("\n\n Produto Escalar (V[i] x K)");

for(i = 0; i < MAX; i++){

printf("\n\t%d - %2d.", i, V[i]\*K);

}

printf("\n\n");

}

1. Faça um programa C que leia um vetor X[20] e após, troque o 1o elemento pelo 20o, o 2o pelo 19o, e assim por diante. Ao final, imprima X.

void main(){

const int MAX = 20;

int X[MAX], i, temp;

*// GERAR VALORES ALEATÓRIOS PARA X[i]*

srand(time(NULL));

for(i = 0; i < MAX; i++){

X[i] = (rand()%50+1);

printf("\n X[%d] = %2d.", i, X[i]);

}

*// INVERTER ELEMENTOS DE X[i]*

for(i = 0; i < MAX/2; i++){

temp = X[i];

X[i] = X[MAX-1-i];

X[MAX-1-i] = temp;

}

*// MOSTRAR VETOR X[i] INVERTIDO*

printf("\n\n");

for(i = 0; i < MAX; i++){

printf("\n X[%d] = %2d.", i, X[i]);

}

printf("\n\n");

}

1. Um vetor é palíndromo se ele não se altera quando as posições dos componentes são invertidas. Por exemplo, o vetor v = {1, 3, 5, 5, 3, 1} é palíndromo. Escreva um programa em C que verifique se um vetor é palíndromo.

void main(){

const int MAX = 6;

int i, v[] = {4, 2, 8, 8, 2, 4};

int ehPal = 1; *// VARIÁVEL BOOLEANA PARA TESTAR SE v[i] É PALÍNDROMO*

*// VERIFICAR SE v[i] É PALINDROMO*

for(i = 0; i < MAX/2; i++){

if(v[i] != v[MAX-1-i])

ehPal = 0;

}

*// MOSTRAR RESULTADO*

if(ehPal)

printf("\n\n v[i] E' PALINDROMO!\n\n");

else

printf("\n\n v[i] NAO E' PALINDROMO!\n\n");

}

1. Faça um programa em C que que leia os valores de dois vetores V1 e V2 com 5 elementos cada. Após, gera um terceiro vetor V, com 10 elementos, intercalando os elementos de V1 e V2. Por exemplo, sendo V1 = { 3, 8, 5, 9, 4} e V2 = { 2, 12, 6, 1, 11}, então V = {3, 2, 8, 12, 5, 6, 9, 1, 4, 11}

void main(){

const int MAX = 5;

int i, V1[MAX], V2[MAX], V3[2\*MAX];

*// GERAR VALORES ALEATÓRIOS PARA V1[i] E V2[i]*

srand(time(NULL));

for(i = 0; i < MAX; i++){

V1[i] = (rand()%11);

V2[i] = (rand()%11);

}

*// GERAR V3[i] INTERCALANDO V1[i] E V2[I]*

for(i = 0; i < MAX; i++){

V3[2\*i] = V1[i];

V3[2\*i+1] = V2[i];

}

*// MOSTRAR OS TRÊS VETORES*

printf("\nV1[] = {");

for(i = 0; i < MAX-1; i++){

printf("%2d, ", V1[i]);

}

printf("%2d}\n", V1[MAX-1]);

printf("\nV2[] = {");

for(i = 0; i < MAX-1; i++){

printf("%2d, ", V2[i]);

}

printf("%2d}\n", V2[MAX-1]);

printf("\nV3[] = {");

for(i = 0; i < 2\*MAX-1; i++){

printf("%2d, ", V3[i]);

}

printf("%2d}\n\n", V3[2\*MAX-1]);

}