

Linguagem C

Professor Alexandre Moreno

VETOR

Um vetor consiste em um arranjo de elementos armazenados, sequencialmente na memória principal, todos acessíveis por meio de um mesmo nome (identificador), ao qual são acessados e referenciados por meio de um índice (LOPES; GARCIA, 2002).

Identificador
do vetor


Índices

notas

0	55
1	72
2	85
3	65
4	80

Em C, os vetores são identificados pela existência de colchetes após o nome da variável no momento da declaração, e o seu primeiro valor estará no endereço 0 (zero)

Primeiro Índice



	notas
0	55
1	72
2	85
3	65
4	80

Declaração do vetor:

```
float notas[5];
```

Qualquer tipo pode ser declarado como um tipo de vetor (int, float, bool, char, etc), entretanto, lembre-se sempre que uma *string* já é um conjunto de caracteres ao qual definimos o seu tamanho (em quantidade de caracteres). Para declararmos um vetor de *strings*, será necessário a definição da quantidade de elementos e também da quantidade de caracteres em cada elemento.

```
char nome[10];
```



Quantidade total de caracteres
na *string* nome

```
char nomes[10][20];
```



Quantidade total de caracteres
na *string* nomes



Quantidade de elementos no vetor

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

int main () {
    char ch;

    int x,n[10];
    for (x=0; x<10; x++){
        printf("digite o %d valor: ",x+1);
        scanf("%d",&n[x]);
    }
    system("pause");
    return 0;
}
```

```
informe o 1o. valor: 15
informe o 2o. valor: 10
informe o 3o. valor: 203
informe o 4o. valor: -8
informe o 5o. valor: 25
informe o 6o. valor: 0
informe o 7o. valor: -68
informe o 8o. valor: 50
informe o 9o. valor: 43
informe o 10o. valor: 21
Pressione qualquer tecla para continuar. . .
```

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


int main () {
    char ch;

    int x,n[10];
    //entrada
    for (x=0; x<10; x++){
        printf("digite elemento %i: ",x+1);
        scanf("%i",&n[x]);
    }
    //saída
    printf("\n");
    for (x=0; x<10; x++)
        printf("%io. elemento: %d\n",x+1,n[x]);
    system("pause");
    return 0;
}
```

```
digite elemento 1: 45
digite elemento 2: 21
digite elemento 3: -30
digite elemento 4: 208
digite elemento 5: 82
digite elemento 6: 0
digite elemento 7: -88
digite elemento 8: 200
digite elemento 9: 7
digite elemento 10: 19
```

```
1o. elemento: 45
2o. elemento: 21
3o. elemento: -30
4o. elemento: 208
5o. elemento: 82
6o. elemento: 0
7o. elemento: -88
8o. elemento: 200
9o. elemento: 7
10o. elemento: 19
```

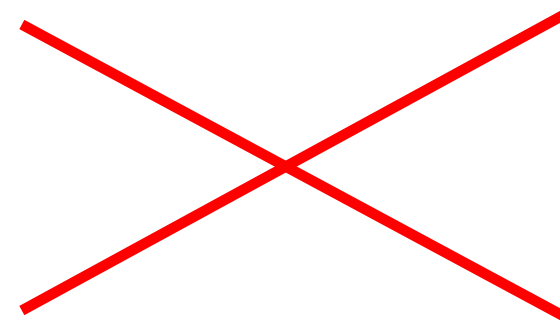
Pressione qualquer tecla para continuar. . .



```
#define n 5
int main () {
    int y = 5;
    float x[n];
    system("pause");
    return 0;
}
```



O valor ou expressão de um array, deve ser sempre inteiro e constante.



Forma Errada:

```
int vetor [y];
int vetor2 [4.5];
```


Atribuição direta

```
int main () {  
    float notas [100];  
    notas [0] = 81; //primeiro índice  
    notas [1] = 55; //segundo índice  
    notas [99] = 62; //último índice  
    system("pause");  
    return 0;  
}
```

Cálculos ou atribuições deverão seguir da mesma forma que variáveis convencionais, apenas lembre-se de utilizar o índice ao mencionar uma variável de vetor.

```
int main () {  
    float notas [100];  
    scanf("%f", &notas[5]); //comando de leitura  
    notas [0] = 10; //comando de atribuição  
    notas [1] = notas [5] + notas [0]; //expressão  
    system("pause");  
    return 0;  
}
```

```
#include <stdio.h>
#include <stdlib.h>
#define n 5
int main () {
    float notas[n], media=0;
    int i;
    for (i = 0; i < n; i++){
        printf("Digite a nota do aluno %d: ", i+1);
        scanf("%f", &notas[i]);
        media = media + notas [i];
    }
    media = media / n;
    printf("\nAlunos acima da media calculada: \n");
    for (i = 0; i < n; i++)
        if (notas [i] > media)
            printf("Aluno %d: %5.2f\n", i, notas[i]);
    system("pause");
}
```

```
Digite a nota do aluno 1: 5
Digite a nota do aluno 2: 7
Digite a nota do aluno 3: 8
Digite a nota do aluno 4: 3
Digite a nota do aluno 5: 6

Alunos acima da media calculada:
Aluno 1:  7.00
Aluno 2:  8.00
Aluno 4:  6.00
Pressione qualquer tecla para continuar.
```

Mostrará apenas alunos com
média superior a 5,8 (29/5)

```
#include <stdio.h>
#include <stdlib.h>
#define n 5
int main () {
    float notas[n], media=0;
    int i;
    for (i = 0; i < n; i++){
        printf("Digite a nota do aluno %d: ", i+1);
        scanf("%f", &notas[i]);
        media = media + notas [i];
    }
    media = media / n;
    printf("\nAlunos acima da media calculada: \n");
    for (i = 0; i < n; i++)
        if (notas [i] > media)
            printf("Aluno %d: %5.2f\n", i, notas[i]);
    system("pause");
}
```

```
Digite a nota do aluno 1: 5
Digite a nota do aluno 2: 7
Digite a nota do aluno 3: 8
Digite a nota do aluno 4: 3
Digite a nota do aluno 5: 6
```

```
Alunos acima da media calculada:
Aluno 1:  7.00
Aluno 2:  8.00
Aluno 4:  6.00
Pressione qualquer tecla para continuar.
```

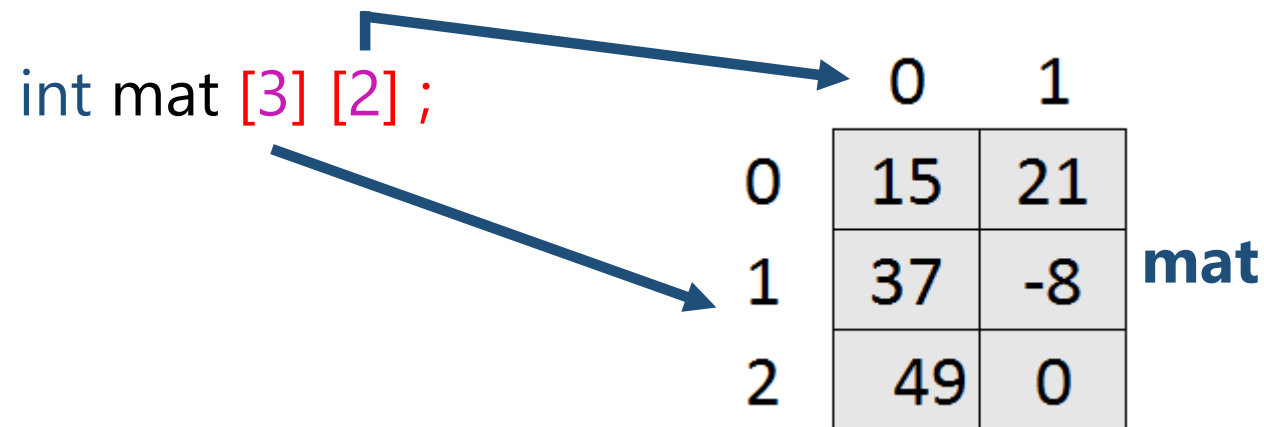
Um detalhe observado é que a quantidade de elementos do vetor pode ser alterada facilmente (inclusive nos cálculos), utilizando uma constante.

MATRIZ

Matriz é uma estrutura de dados multidimensional (2 ou mais dimensões), identificada por um nome e tipo (int, float, char, bool, etc)

tipo nome [tam1] [tam2];

//exemplo: para guardar 6 informações em uma matriz



Podemos visualizar a dimensão da matriz com 3 linhas e 2 colunas

```
int mat [3] [2] ;
```

■ `mat [0] [0] = 5 ;` //primeira posição

■ `mat [0] [1] = 23;`

■ `mat [2] [1] = -1 ;` //última posição

mat		
	0	1
0	5	23
1		
2		-1

Um exemplo de uso de matriz:

Solicitar 2 notas de uma turma contendo 10 alunos, calcular a média e armazenar todos os dados em uma matriz.

	0	1	2
0	7,0	4,0	5,5
1	8,0	6,0	7,0
2	0,0	5,0	2,5
3	4,0	6,0	5,0
4	7,0	6,0	6,5
5	7,0	7,0	7,0
6	8,0	4,0	6,0
7	9,0	5,0	7,0
8	6,0	7,0	6,5
9	5,0	6,0	5,5

```
float notas[10][3];
```



```
#include <stdio.h>
#include <stdlib.h>
int main () {
    float media, notas[10][3];
    int x,y;
    for (x = 0; x < 10; x++){
        media=0;
        printf("\nAluno: %d\n", x+1);
        for (y=0; y<2; y++){
            printf("Informe a nota %i: ", y+1);
            scanf("%f", &notas[x][y]);
            media=media+notas[x][y];
        }
        notas[x][2]=media/2;
    }
    system("pause");
}
```

```
Aluno: 1
Informe a nota 1: 7
Informe a nota 2: 8
```

```
Aluno: 2
Informe a nota 1: 9
Informe a nota 2: 8
```

```
Aluno: 3
Informe a nota 1: 5
Informe a nota 2: 7
```

```
Aluno: 4
Informe a nota 1: 0
Informe a nota 2: 8
```

```
Aluno: 5
Informe a nota 1: 5
Informe a nota 2: 9
```

Um exemplo de uso de matriz:

Solicitar **o nome** e 2 notas de uma turma contendo 10 alunos, calcular a média e armazenar todos os dados em uma matriz.

		0	1	2
0	JOSE CARLOS	7,0	4,0	5,5
1	MARIA EDUARDA	8,0	6,0	7,0
2	CARLOS ANDRADE	0,0	5,0	2,5
3	ALEXANDRE SILVESTRE	4,0	6,0	5,0
4	PAULA VALENTINA	7,0	6,0	6,5
5	BEATRIZ NOGUEIRA	7,0	7,0	7,0
6	TADEU ROMEU JUNIOR	8,0	4,0	6,0
7	MARCELO CAMARGO	9,0	5,0	7,0
8	ALBERTO SOARES	6,0	7,0	6,5
9	NAIANE FERRAREZI	5,0	6,0	5,5

```
char nomes[10][50];  
float notas[10][3];
```

Um exemplo de uso de vetor e matriz:

Solicitar **o nome** e 2 notas de uma turma contendo 10 alunos, calcular a média e armazenar todos os dados em uma matriz.

		0	1	2
0	JOSE CARLOS	7,0	4,0	5,5
1	MARIA EDUARDA	8,0	6,0	7,0
2	CARLOS ANDRADE	0,0	5,0	2,5
3	ALEXANDRE SILVESTRE	4,0	6,0	5,0
4	PAULA VALENTINA	7,0	6,0	6,5
5	BEATRIZ NOGUEIRA	7,0	7,0	7,0
6	TADEU ROMEU JUNIOR	8,0	4,0	6,0
7	MARCELO CAMARGO	9,0	5,0	7,0
8	ALBERTO SOARES	6,0	7,0	6,5
9	NAIANE FERRAREZI	5,0	6,0	5,5

```
char nomes[10][50];  
float notas[10][3];
```

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    char nomes[10][50];
    float media, notas[10][3];
    int x, y;
    for (x = 0; x < 10; x++) {
        media=0;
        printf("\nAluno%d: ", x+1);
        gets(nomes[x]);
        for (y=0; y<2; y++) {
            printf("Informe a nota %i: ", y+1);
            scanf("%f", &notas[x][y]);
            fflush(stdin);
            media=media+notas[x][y];
        }
        notas[x][2]=media/2;
    }
}
```

```
Aluno1: JOSE CARLOS
Informe a nota 1: 7
Informe a nota 2: 4

Aluno2: MARIA EDUARDA
Informe a nota 1: 8
Informe a nota 2: 6

Aluno3: CARLOS ALEXANDRE
Informe a nota 1: 0
Informe a nota 2: 5

Aluno4: PAULA VALENTINA
Informe a nota 1: 7
Informe a nota 2: 6

Aluno5: BEATRIZ NOGUEIRA
Informe a nota 1: 7
Informe a nota 2: 7

Aluno6: TADEU ROMEU JUNIOR
Informe a nota 1: 8
Informe a nota 2: 4
```

Uma matriz pode conter:

```
int m1 [5]; // 1 dimensão
```

```
int m2 [3] [2] ; // 2 dimensões
```

```
int m3 [5] [3] [6] ; // 3 dimensões
```

```
int m [6] [2] [3] [4] ; // 4 dimensões ou mais
```

Inicialização de Array

Inicializando um array:

```
// preenche o array com um valor definido
```

```
int vetor [5];
```

```
int i;
```

```
for (i = 0; i < 5; i++){
```

```
    vetor [i] = 0;
```

```
}
```

Inicializando um array:

// preenche o array com valores diferentes em cada posição

int vetor [5];

vetor [0] = 50;

vetor [1] = 40;

vetor [2] = 30;

vetor [3] = 20;

vetor [4] = 10;

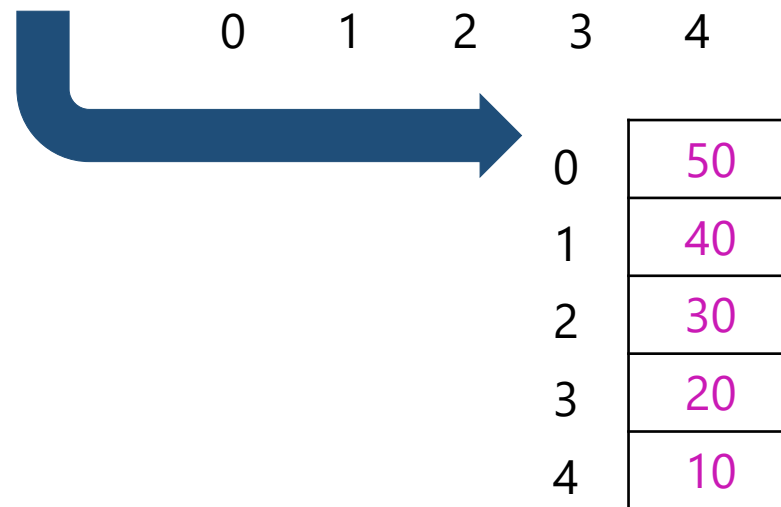
Este exemplo funcionaria muito bem com um array de poucas posições. Imagina um array de 40 posições?



Podemos definir uma lista de valores para serem atribuídos no array:

`tipo nome_array [tam1] [tam2] = {valores};`

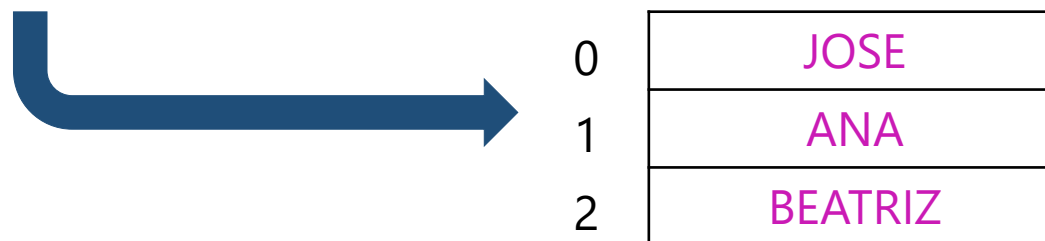
`int vetor [5] = {50, 40, 30, 20, 10};`



Podemos definir uma lista de valores para serem atribuídos no array:

```
tipo nome_array [tam1] [tam2] = {valores};
```

```
char nomes [3] = {"JOSE", "ANA", "BEATRIZ"};
```



// inicializando um vetor

```
float vetor [3] = {1.5, 10.1, 4.25};
```

// inicializando uma matriz

```
int matriz [2][3] = {1, 2, 3, 4, 5, 6};
```

Coluna 0, 1, 2

```
int matriz [2][3] = {{1, 2, 3}, {4, 5, 6}};
```

Linha 0

Linha 1

Inicialização sem o tamanho do array declarado

// este exemplo, declara um array com 10 posições

```
int vetor [ ] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

// o número de linhas da matriz será 5

```
int matriz [][][2] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
```

Maior valor de um Array

Encontrando o maior elemento de um array:

```
int i, lista [5] = {3, 18, 2, 51, 45};  
int maior = lista [0];  
for (i = 1; i < 5; i++){  
    if (maior < lista [i]){  
        maior = lista [i];  
    }  
}  
printf("Maior = %d\n", maior);
```

Encontrando o maior elemento de uma matriz:

```
int mat [3] [3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
int i, j, maior = mat [0] [0];  
for (i = 1; i < 3; i++){  
    for (j = 0; j < 3; j++){  
        if (maior < mat [i] [j])  
            maior = mat [i] [j];  
    }  
}  
printf("Maior = %d\n", maior);
```

Somando os elementos de um Array

Somando os elementos de um array

```
int i, lista [5] = {1, 2, 3, 4, 5};  
int soma = 0;  
for (i = 0; i < 5; i++){  
    soma = soma + lista [i];  
}  
printf("Soma = %d\n", soma);
```

Somando os elementos de uma matriz:

```
int mat [3] [3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};  
int i, j, soma = 0;  
for (i = 0; i < 3; i++){  
    for (j = 0; j < 3; j++){  
        soma = soma + mat [i] [j];  
    }  
}  
printf("Soma = %d\n", soma);
```

Exemplos

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

char frase[50];
char vogal[11]="aeiouAEIOU";
int x,y,tam,contavog=0;
int main() {
    printf("Informe uma frase ");
    gets(frase);
    tam=strlen(frase);
    for (x=0; x<tam; x++)
        for (y=0; y<10; y++)
            if (frase[x]==vogal[y])
                contavog++;
    printf("número de vogais: %d\n\n",contavog);
    system("pause");
    return(0);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char nome[3][50];
int x;
int main() {
    //entrada
    for (x=0; x<3; x++) {
        printf("Nome %d: ", x+1);
        gets(nome[x]);
    }
    printf("\n");
    //saída
    for (x=0; x<3; x++) {
        printf("Nome %d: %s \n", x+1, nome[x]);
    }
    system("pause");
    return(0);
}
```

```
char nome[3][4][50];
int x,y;
int main() {
    //entrada
    for (x=0; x<3; x++) {
        for (y=0; y<4; y++) {
            printf("Nome %d x %d: ",x+1,y+1);
            gets(nome[x][y]);
        }
        printf("\n");
    }
    //saída
    for (x=0; x<3; x++) {
        for (y=0; y<4; y++) {
            printf("Nome %d x %d: %s \n",x+1,y+1,nome[x][y]);
        }
        printf("\n");
    }
    system("pause");
    return(0);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
int num[10];
int x, y;
int main() {
    srand(time(NULL));
    //entrada
    for (x=0; x<10; x++) {
        num[x]=rand()%10;
    }
    //saida
    for (x=0; x<10; x++) {
        printf("    %d\n", num[x]);
    }
    system("pause");
    return(0);
}
```

Construa um programa utilizando vetores e matrizes, que armazene os dados de uma turma contendo o nome do aluno, 2 notas bimestrais, 1 nota substitutiva, média final, a quantidade de faltas e a situação do aluno (“Aprovado” ou “Reprovado”).

A turma terá no máximo 40 alunos e deverá ser solicitado a quantidade de alunos na sala e também o nome e carga horária da disciplina antes de solicitar e calcular os dados dos alunos.

Para aprovação, o aluno deverá ter no mínimo média 6 e frequência de 75%

Os dados de entrada deverão ser validados.

No final, apresente todos os dados dos alunos.

Obs. Pergunte ao usuário se ele quer digitar os dados ou gerar valores randômicos (inclusive para o nome)

 H:\CESUMAR\DEV_CONIO\matriz1.exe

```
Disciplina: Algoritmos
Carga Horaria: 80
Quant. Alunos (max 40): 20
(D)igitar ou dados ou (S)ortear dados? _
```

```
----- Aluno 1 -----
Nome.....: MARIA SILVA
      Nota 1: 6
      Nota 2: 7
Substitutiva: 4
Frequencia: 60

----- Aluno 2 -----
Nome.....: CARLOS ANTONIO
      Nota 1: 4
      Nota 2: 3
Substitutiva: 9
Frequencia: 78

----- Aluno 3 -----
Nome.....: MARIA IZABEL
      Nota 1: 3
      Nota 2: 9
Substitutiva: 6
Frequencia: 75
```

Usuário optou por
Digitar os dados ao
invés de **S**ortear.

Disciplina: Algoritmos II - Carga Horaria: 80 Quant. Alunos: 20

nome	nota1	nota2	subst	media	frequencia	situacao
JOAO CARLOS	10.00	0.00	6.00	8.00	54 (68%)	Reprovado
EDSON MORENO	3.00	8.00	4.00	6.00	18 (23%)	Reprovado
ALEXANDRE RUIZ	4.00	9.00	10.00	9.50	75 (94%)	Aprovado
PAULA LORENZONI	0.00	2.00	2.00	2.00	6 (8%)	Reprovado
BEATRIZ SILVA	1.00	3.00	2.00	2.50	0 (0%)	Reprovado
DIONIZIO COSTA	3.00	3.00	3.00	3.00	53 (66%)	Reprovado
LUIZA SHIGUITI	9.00	3.00	8.00	8.50	75 (94%)	Aprovado
PAULA LORENZONI	10.00	0.00	3.00	6.50	73 (91%)	Aprovado
ROBSON GALVAO	10.00	10.00	9.00	10.00	24 (30%)	Reprovado
CRISTIANE MARA	1.00	8.00	6.00	7.00	63 (79%)	Aprovado
ALEXANDRE RUIZ	9.00	1.00	4.00	6.50	2 (3%)	Reprovado
CRISTIANE MARA	0.00	2.00	1.00	1.50	61 (76%)	Reprovado
LUCIA BANZ	6.00	10.00	6.00	8.00	70 (88%)	Aprovado
GUSTAVO COSTA	5.00	5.00	3.00	5.00	6 (8%)	Reprovado
CARLOS CRISTIANO	6.00	10.00	1.00	8.00	46 (58%)	Reprovado
TOMAZ THIENGO	8.00	9.00	4.00	8.50	6 (8%)	Reprovado
JANAINA LIZ	2.00	1.00	7.00	4.50	53 (66%)	Reprovado
ANA LUCIA	6.00	5.00	6.00	6.00	78 (98%)	Aprovado
ROBSON GALVAO	4.00	5.00	9.00	7.00	65 (81%)	Aprovado
LILIAN DOMINGUES	9.00	3.00	6.00	7.50	48 (60%)	Reprovado

Tela de resultado dos dados

Pressione qualquer tecla para continuar. . .