



1 Assunto

Vamos estudar como juntar muitas variáveis de um mesmo tipo em uma forma conveniente, conhecida como vetor ou arranjo.

2 Vetores: O Que São e Para Que Servem

O seguinte programa lê a nota de cinco alunos, calcula a média e mostra na tela:

```
#include <stdio.h>

int main()
{
    float nota1, nota2, nota3, nota4, nota5, media;

    printf("Digite a nota do aluno 0: ");
    scanf("%f", &nota1);
    printf("Digite a nota do aluno 1: ");
    scanf("%f", &nota2);
    printf("Digite a nota do aluno 2: ");
    scanf("%f", &nota3);
    printf("Digite a nota do aluno 3: ");
    scanf("%f", &nota4);
    printf("Digite a nota do aluno 4: ");
    scanf("%f", &nota5);
    media = (nota1 + nota2 + nota3 + nota4 + nota5) / 5;
    printf("A média é: %f\n", media);

    return 0;
}
```

Imagine agora que você quer fazer o mesmo com a nota de 1000 alunos. Os dois grandes inconvenientes são ter que nomear 1000 variáveis e não poder usar o `for`¹ já que temos que especificar cada variável para o `scanf`.

Vetor em C, é uma forma de agrupar e nomear facilmente muitas variáveis do mesmo tipo. Se quisermos 1000 variáveis do tipo `float` organizadas em um vetor, fazemos:

¹Na verdade, pode, mas de uma forma pouco recomendada.

```
TIPO IDENTIFICADOR[QUANTIDADE];
```

Onde TIPO pode ser qualquer um dos tipos básicos do C (`float`, `int`, `char`, `unsigned`, etc...), IDENTIFICADOR é o nome do vetor e QUANTIDADE diz o número de elementos que o vetor deve poder armazenar. Então, se quisermos armazenar 1000 notas, podemos escrever:

```
float nota[1000];
```

A primeira nota é `nota[0]` e a última é `nota[999]`. Isto é muito importante! A nésima nota é acessada com `nota[n-1]` e usamos **colchetes** e não parênteses em vetores. Parênteses são para funções apenas. Como você deve ter percebido também, o que estudamos anteriormente como cadeias de caracteres, nada mais são que vetores do tipo `char`.

O programa a seguir lê 1000 notas, depois calcula a média e mostra na tela:

```
#include <stdio.h>

int main()
{
    int i, quantidade;
    float nota[1000], media, soma=0;

    for (i = 0; i < 1000; ++i) {
        printf("Digite a nota do aluno %d: ", i);
        scanf("%f", &nota[i]);
        soma += nota[i];
    }

    media = soma / 1000;
    printf("A média é: %f\n", media);

    return 0;
}
```

Perceba como é mais prático usar vetores neste caso. Acessar os elementos do vetor é semelhante a acessar uma variável comum, a única diferença é que temos que especificar um índice entre colchetes. Este índice pode ser um número, uma variável ou uma expressão que tenha como resultado um número inteiro, por exemplo:

```
#include <stdio.h>
#define TAMANHO 11

int main()
{
    float dado[TAMANHO], media3p, soma;
```

```

int i, passo;

// Lê os dados da entrada padrão
for (i = 0; i < TAMANHO; ++i)
    scanf("%f", &dado[i]);

// Mostra a média dos três primeiros
media3p = (dado[0] + dado[1] + dado[2]) / 3;
printf("A média dos três primeiros elementos é: %.2f\n", media3p);

// Mostra o valor do elemento central (neste caso dado[5])
printf("O valor do elemento central é: %f\n", dado[TAMANHO / 2]);

// Somatório
for (i = 0, soma = 0; i < TAMANHO; ++i)
    soma += dado[i];

// Mostra elementos na mesma ordem em que foram digitados,
// e na ordem inversa
for (i = 0; i < TAMANHO; ++i) //
    printf("%6.2f\t%6.2f\n", dado[i], dado[TAMANHO-1 - i]);

return 0;
}

```

2.1 Lendo um Número Desconhecido de Elementos

As vezes queremos armazenar um número variável de valores. Por exemplo, se sabemos que o máximo de alunos matriculados numa disciplina é 40, podemos definir um vetor com tamanho suficiente para armazenar todos e usar apenas aqueles que forem necessários. O programa a seguir lê as notas de no máximo 40 alunos. O programa para de ler as notas caso o usuário digite um número negativo, ou o limite de 40 seja alcançado.

```

#include <stdio.h>
#define LIM 40

int main(){
    float nota[LIM], soma = 0.0;
    int i = 0;

    do {
        if(i >= LIM){
            printf("Limite alcançado.\n");

```

```

        i++;
        break;
    }
    printf("Digite a nota do aluno %d: ", i);
    scanf("%f", &nota[i]);
    if(nota[i] >= 0)
        soma += nota[i];
} while (nota[i++] >= 0);

printf("A média das notas é: %.2f\n", soma / (i - 1));

return 0;
}

```

2.2 Inicialização do Vetor

Vetores, como variáveis, podem ser inicializados assim que são declarados. Por exemplo:

```

#include <stdio.h>

int main(){
    int moedas[5] = {50, 25, 10, 5, 1};
    int i;

    for (i = 0; i < 5; ++i)
        printf("%d ", moedas[i]);
    printf("\n");

    return 0;
}

```

mostra na tela:

```
50 25 10 5 1
```

2.3 Um Pouco Mais Sobre Cadeias de Caracteres

Como vimos, cadeias de caracteres, ou **strings**, são vetores de caracteres. Um problema com a função **fgets** é que ela armazena também o '**\n**'. Para eliminar o '**\n**' basta percorrer o vetor, e colocar um '**\0**' (delimitador de fim de cadeia) no lugar do '**\n**', da seguinte forma:

```

#include <stdio.h>

int main(){

```

```

char nome[50];

fgets(nome, 50, stdin);

for (i = 0; i < 50; ++i)
    if (nome[i] == '\n') {
        nome[i] = '\0';
        break;
    }

printf("%s.\n", nome);

return 0;
}

```

Vetores, ou cadeias, de caracteres podem ser inicializados da mesma forma que para números. Por exemplo, poderíamos ter uma *string* inicializada da seguinte forma:

```
char mensagem[] = {'O', 'i', ' ', 'm', 'u', 'n', 'd', 'o', '!', '\0'};
```

O '\0' no fim da cadeia é o caractere que marca o fim da cadeia, como já estudamos, toda a cadeia tem um deste no fim. Obviamente, é um tanto quanto chato escrever desta forma, por isto, C oferece um “atalho”, ou sintaxe alternativa, para escrever a mesma coisa. É justamente a forma que já conhecemos:

```
char mensagem[] = "Oi mundo!";
```

2.4 Notas Acima da Média

Podemos modificar o programa das notas anterior para depois de mostrar a média das notas, mostrar todos os alunos com notas acima da média, junto com as médias.

```

#include <stdio.h>
#define LIM 30

int main(){
    float nota[LIM], soma, media;
    int i, quantidade;

    for (i = 0, soma = 0; i < LIM; ++i) {
        printf("Digite a nota do aluno %d: ", i);
        scanf("%f", &nota[i]);

        if (nota[i] < 0) {
            i++;
        }
    }

    media = soma / LIM;
    printf("A média é %.2f\n", media);

    for (i = 0; i < LIM; ++i) {
        if (nota[i] > media) {
            printf("Nota %.2f é maior que a média (%.2f)\n", nota[i], media);
        }
    }
}

```

```
        break;
    }

    soma += nota[i];
}

quantidade = i - 1;
media = quantidade > 0 ? soma / quantidade : 0;

printf("A média das notas é: %.2f\n", media);

for (i = 0; i < quantidade; ++i) {
    if (nota[i] >= media)
        printf("Aluno %d está acima da média: %.2f\n", i, nota[i]);
}

return 0;
}
```