Vetores e Funções

Já conhecemos as funções, já conhecemos os vetores. Agora, é hora de fazermos uso dos vetores em funções. Show, vamos lá!

No último material vimos que para um vetor x de tamanho 5, conforme representação a seguir:

int x[5] = {7,0,3,9};

O acesso ao seu primeiro elemento seria através de x[0] ou x, pois do endereço do primeiro elemento do vetor é a mesma coisa de falar apenas o nome da variável que representa esse vetor. E desta forma, por tratar-se de endereços, poderíamos acessar o conteúdo do endereço do primeiro elemento de x por *x. E verificamos que:

> Vetores são conjuntos de endereços alocados de maneira homogênea no espaço de memória.

Desta maneira, podemos manipulá-los por meio de ponteiros. Então todas as operações que podemos realizar com ponteiros podemos aplicar na manipulação de vetores.

Com ponteiros	É equivalente a
*x	x[0]
x	&x[0]
x[índice]	*(índice)
&x[índice]	(x+ índice)

Sabendo disso, como declarar uma função que possui como parâmetro um vetor?

Declaração

Existem duas formas para declarar uma função cujo um dos parâmetros seja um vetor. Para exemplificar, imaginemos uma função com o objetivo de contar quantos números são positivos em um vetor (v) de números inteiros, sua assinatura (ou protótipo) poderia dado por

```
int contagem_positiva (int v[], int tamanho);
ou
int contagem_positiva (int* v, int tamanho);
```

Como vimos anteriormente para termos acesso ao primeiro elemento de um vetor podemos fazê-lo através de ponteiros, assim, na passagem de um vetor para uma função podermos tanto fazer por **v[ ]** como também por ***v**.

Observe também, essa função não sabe qual o tamanho do vetor, desta maneira precisará receber como parâmetro também o número de elementos que o vetor possui, sendo assim, seu uso é mais flexível.

Uso

O uso é simples, usando a declaração de um vetor como parâmetro:

```
int contagem_positiva (int v[], int tamanho){
        int i;
        int cont=0;

        for(i=0; i<tamanho;i++)
                if(v[i] > 0)
                        cont++;
        return cont;
}
#define TAM 5
int main()
{
        int vetor[TAM]={-2,3,-1,0,4};
        int positivos=0;

        positivos = contagem_positiva(vetor, TAM);

        printf("%d", positivos);
}
```

Indicação de um vetor como parâmetros

Definição de uma constante

Como vetores são endereços de memórias homogêneas, não é preciso passar &v[0] para indicar o endereço do primeiro elemento do vetor, basta indicar seu nome

Usando a declaração de um vetor como um ponteiro:

```
int contagem_positiva (int* v, int tamanho){
        int i;
        int cont=0;

        for(i=0; i<tamanho;i++)
                if(v[i] > 0)
                        cont++;
        return cont;
}
#define TAM 5
int main()
{
        int vetor[TAM]={-2,3,-1,0,4};
        int positivos=0;

        positivos = contagem_positiva(vetor, TAM);

        printf("%d", positivos);
}
```

Indicação de um ponteiro como parâmetro

Algumas Curiosidades

Bom, algumas perguntas podem surgir:

1) Posso passar na declaração da função o tamanho do vetor? Fazendo uso do exemplo anterior, algo como:

```
int contagem_positiva (int v[5], int tamanho);
ou
int contagem_positiva (int v[TAM], int tamanho);
```

*Sim, é possível. Porém é irrelevante na declaração da função. Por isso, habitualmente encontramos assinaturas com formato **v[]** ou ***v**.*

2) Devo sempre indicar o tamanho do vetor como um parâmetro da função? Como no exemplo anterior:

```
int contagem_positiva (int v[], int tamanho);
ou
int contagem_positiva (int* v, int tamanho);
```

Como já aprendemos, as funções, na programação, possuem a responsabilidade de modularizar nosso código, de organizar, de ser acessível e flexível a diversas situações. Enviar o tamanho do vetor como um parâmetro a mais na função permite que ela seja aplicável a qualquer vetor, do mesmo tipo da declaração, independente do seu tamanho.

Super legal, né? Então, vamos praticar, exercitar e trocar várias ideias no nosso fórum.