

Aulão de IPC

Anghie

Junho - 2024

Conteúdo

1	Funções	2
2	Vetores	4
2.1	Definição	4
3	Matriz	5
3.1	Definição	5
4	Struct	6
4.1	Definição	6
5	Alocação Dinâmica	7
5.1	Definição	7

1 Funções

Definição

Uma função em programação não é tão diferente de uma função na matemática, ou seja, uma função na programação recebe um valor de um tipo, e devolve um valor em algum tipo. Mas o que uma função faz em si? Matematicamente falando, seja f uma função que pega um valor em \mathbb{N} e retorna um valor em \mathbb{Z} , ou seja:

$$f(x) = x * (-1)^x$$

Em resumo, a função acima pega um $x \in \mathbb{N}$ qualquer que se for par vai ser igual a ele mesmo, e se for ímpar vai ser igual ao oposto dele.

Mas o que isso tem haver com programação? Vamos criar a mesma função, mas em C:

Exemplo

```
1  #include <stdio.h>
2
3  int ParOuImpar(int x){
4      if(x % 2 == 0){
5          x = x;
6      } else {
7          x = x * (-1);
8      }
9
10     return x;
11 }
12
13 int main(){
14     int x;
15     scanf("%d", &x);
16
17     x = ParOuImpar(x);
18
19     printf(x);
20
21     return 0;
22 }
```

Figura 1: Nossa função em C

Como vocês puderam ver, nosso código é um código simples, pois ele vai pedir ao usuário um inteiro qualquer, e vai falar que o valor desse inteiro será o resultado da nossa função. E o que nossa função faz?

Bom, note que nossa função é do tipo inteiro, ou seja, no final da execução dela, no return terá um valor do tipo *int*, e o que ela recebe? Dentro do paranteses da nossa função está escrito *int x*, o que significa que ela recebe um valor qualquer chamado de x que é do tipo inteiro. E como ela funciona? Bom, ela verá se o nosso x em questão é par, e retornará o próprio x se for verdade ou o oposto dele, se x for ímpar.

2 Vetores

2.1 Definição

Todos nós sabemos o que é um vetor graças a GA, mas e na nossa querida programação? Bom, uma variável que é um vetor é nada mais nada menos que uma variável que pode guardar diversos valores, portanto, um vetor em C é basicamente um conjunto. Primeiramente, nos lembremos do que é um conjunto:

Um conjunto na matemática é, de maneira simplória, um conjunto é uma junção de números, por exemplo, os \mathbb{N} são todos os números maiores ou iguais a 0 sem uma parte racional. Mas a principal e mais importante diferença entre um vetor e um conjunto na programação é que um vetor pode ser de diferentes tipos, como *char*, *int*, *float* etc.

Na prática um vetor é uma forma mais fácil de se organizar múltiplas variáveis do mesmo tipo, para criar um vetor, primeiro falamos para o computador o nome dele, depois colocamos entre [] o tamanho que desejamos para o nosso vetor, para facilitar o entendimento, vamos a um exemplo:

Exemplo

```
19 char nome[5];
20 int numerosFavoritos[4];
21 printf("Digite seu nome:\n>>> ");
22 scanf(" %s", &nome);
23
24 printf("Digite seus 4 numeros favoritos:\n>>> ");
25 for(int i = 0; i < 4; i++){
26     scanf("%d", &numerosFavoritos[i]);
27 }
28
29 printf("%s", nome);
30 for(int i = 0; i < 4; i++){
31     printf("%d", numerosFavoritos[i]);
32 }
```

Figura 2: Vetor em C

Como é possível observar, temos 2 diferentes vetores em nosso exemplo, e eles foram criados para receber os valores: "Pedro", 2, 4, 22 e 13. Note que o número de elementos é menor do que o número que coloquei nos vetores, temos apenas 5 elementos, mas temos 9 elementos no total nos vetores. Isso acontece porque no C, uma *string* é tratada como um vetor de *char*. Ainda não vimos isso, mas uma *string* é uma linha de texto com mais de um caractere. Ou seja, o valor "Pedro" é na verdade um vetor de 5 caracteres: ['P','e','d','r','o'].

Vale a pena saber o conceito de uma *string*, apesar dela não existir de fato em C por conta do *%s*, isso diz para o nosso querido *scanf* que ele vai receber um valor do tipo *string*, o que permite o não uso do *for*, como é visto para preencher nosso vetor de números favoritos, e ele também funciona da mesma forma para o *printf*.

Vale ressaltar que para usar um vetor em C, temos que nos lembrar que ele guarda item a item usando um dígito como identificador. Ou seja, em nosso vetor de números favoritos ele terá como identificadores todos os naturais em [0, 4[ou seja, 0, 1, 2e3, nesta mesma ordem. Isso está sendo usado nos *for* acima, onde o valor de *i* também é todos os naturais no nosso intervalo de [0, 4[, em ordem crescente. Isso também explica como usamos um vetor. Visto que normalmente vamos manipular um elemento por vez nos vetores, para especificar qual, digitamos o nome do vetor seguido de [] com o índice desejado dentro.

3 Matriz

3.1 Definição

Uma matriz é praticamente a mesma coisa que uma matriz matemática, porém segue as mesmas regras que nossos vetores. Ou seja, para criar uma, e preciso nomeá-la e colocar os `[]` com o tamanho da matriz, porém ela recebe dois `[]`, visto que uma matriz tem no mínimo duas dimensões.

Exemplo

Vamos fazer um jogo da velha simples, de uma jogada só, onde o 'X' deve ganhar, para isso, montaremos uma matriz quadrada de grau 3, ou seja, uma matriz 3x3:

```
36 char jogoDaVelha[3][3] = {
37     {'X', ' ', 'X'},
38     {'O', 'O', ' '},
39     {' ', 'O', ' '}
40 };
41
42 for(int i = 0; i < 3; i++){
43     for(int j = 0; j < 3; j++){
44         if(j == 0){
45             printf(" %c|", jogoDaVelha[i][j]);
46         }else if(j == 2){
47             printf("%c ", jogoDaVelha[i][j]);
48         } else {
49             printf("%c|", jogoDaVelha[i][j]);
50         }
51     }
52     if(i == 2){
53         printf("\n");
54     } else {
55         printf("\n_____ \n");
56     }
57 }
58 printf("Vez do 'X', ganhe o jogo: (x, y)\n>>> ");
59 int x, y;
60 scanf("%d %d", &x, &y);
61 x--;
62 y--;
63
64 jogoDaVelha[y][x] = 'X';
65
```

Figura 3: Matriz em C

Note que para poder imprimir uma matriz, precisamos de dois *for*, visto que igual na matemática, os índices dos elementos de uma matriz são dados pelo formato ij . Também observe que estamos agora pedindo para o usuário uma coordenada (x, y) para que possamos mudar algum elemento em específico da matriz.

Também é preciso notar duas coisas, para acessar um item dentro de uma matriz, fazemos um processo parecido com o do vetor, dizemos o nome da matriz acompanhado de dois `[]` com os índices ji , isso mesmo, de forma invertida.

4 Struct

4.1 Definição

Exemplo

5 Alocação Dinâmica

5.1 Definição

Exemplo