Funções

Funções em C

Quando queremos resolver um problema, em geral tentamos dividi-lo em *subproblemas* mais simples e relativamente independentes, e resolvemos os problemas mais simples um a um. Uma função cria uma maneira conveniente de encapsular alguns detalhes de "processamento", ou seja, como algum resultado é obtido.

Criando funções, um programa C pode ser estruturado em partes relativamente independentes que correspondem as subdivisões do problema. Funções como: `toupper()`, `sqrt()` , são funções de uma biblioteca padrão (do C). Você não sabe como elas foram escritas, mas pode saber como utilizá-las.

Ou seja, você sabe o nome das funções e quais informações específicas você deve fornecer a elas (valores que devem ser passados para as funções) para que a função produza os resultados esperados. Quando nos referirmos a uma função neste texto usaremos a maneira frequentemente utilizada que é o nome da função seguido de () .

As funções – como variáveis – requerem nomes; as mesmas regras aplicáveis aos nomes de variáveis também se aplicam a nomes de funções (sequência de letras ou dígitos numéricos, mas não iniciando com um dígito; nenhum caractere especial, exceto o sublinhado).

Funções em C

Tomemos como exemplo o programa abaixo, que nada mais é um "Hello, World!" usando conceito de funções:

```c
#include <stdio.h>

// criamos a função
void minha_funcao(){
    printf("Olá, mundo!\n");
}

main(){
    // chamamos a função
    minha_funcao();
}
```

Tipos de Funções em C

O tipo de uma função está diretamente relacionado ao tipo de retorno da função. Funções tipo void não tem e não devem ter retorno.

```
void minha_funcao(){
    printf("Eu sou uma função\n");
    return 0; // erro
}
```

Mas todos os tipos que podemos atribuir à variáveis também são atribuíveis à funções: int, float, bool e void .

As mesmas regras para criação de nomes de variáveis se aplicam a criação para nomes de função.

Tipos de Funções em C

Parâmetros de funções são valores que devem ser passados para ela.

```c
#include <stdio.h>

// criamos a função
int soma( int x, int y ){
    return x + y;
}

int main(){
    // chamamos a função
    printf("%d\n”, soma(3, 6));
    return 0;
}
```

Tipos de Funções em C

Funções da biblioteca padrão: STD(Standard Library) do C. Vamos conhecer algumas como:

Converter para MAIÚSCULA: toupper()

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

int main(){
  char a = 'a';

  printf("%c\n", toupper( a ));

  a = toupper( a );
  printf("%c\n", a );

  return 0;
}
```

E entre outras que veremos ao decorrer do curso e quando vermos sobre cabeçalhos.

Tipos de Funções em C

Protótipos de funções, são utilizados para facilitarmos o trabalho do compilador e o mesmo gerar um binário sem "*buracos*" .

Código binário gerado COM "buracos":

```
01010000 01110010 01101001       01101101 01100101      01101001 01110010 01101111
       01010100 01100101        01110010 01100011 01100101 01101001 01110010
                        01101111
```

Código binário gerado SEM "*buracos*":

```
01010000 01110010 01101001 01101101 01100101 01101001 01110010 01101111
01010100 01100101 01110010 01100011 01100101 01101001 01110010 01101111
```

Essa prática também é interessante para questão de endereçamento de memória. É mais fácil encontrar o endereço da função dessa forma.

Tipos de Funções em C

Exemplo de uma função com protótipo:

```c
#include <stdio.h>

int produto(int x,int y);

int main(){
  printf("O produto de 3 x 6 é: %d\n", produto(3, 6));
  return 0;
}

int produto( int x, int y ){
  return x * y;
}
```

Um dica de boa prática é indicar somente o tipo das variáveis passadas.