

UEMG - Universidade do Estado de Minas Gerais - Unidade Frutal

Curso - Sistemas de Informação, 1º Período, Noturno

Disciplina - Introdução à Programação

Docente - Profº Sergio Carlos Portari Junior

Discente - Carlos Augusto de Oliveira Alves

Trabalho - Trabalho de Funções

Data - 24/08/2021

Complementos de Função em C

Na linguagem C, função é um conjunto de comandos que realiza uma tarefa específica em um módulo dependente de código. Simplificando, a função é uma **subrotina** dentro de um programa.

A utilização de uma função em um programa objetiva modular o programa, criando uma estrutura organizada para o programa. Fazendo isso, nosso programa pode ser dividido em várias funções bem definidas, para serem usadas no “main()”.

- **Exemplo de uma Função de multiplicação entre dois números:**

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

```
int mult(int n1, int n2)
```

```
//mult recebe n1e n2 e retorna um int
```

```
{
```

```
    int result;
```

```
    result = n1 * n2;
```

```
    return(result);
```

```
    //retorna o valor para o programa main
```

```
}
```

```
int main(void)
```

```
{
```

```
int v1, v2, result;
printf("Digite o primeiro valor:");
scanf("%d", &v1);
printf("Digite o segundo valor:");
scanf("%d", &v2);
//chama a função e recebe o retorno
result = mult(v1,v2);
printf("Resultado = %d\n", result);
return 0;
}
```

Protótipos de funções em C

Normalmente, as funções são escritas antes do “main()”, mas muitas vezes, para simplificar a leitura de uma função pelo compilador, temos de criar um protótipo de função. Protótipos de Funções são declarações de funções, isto é, você declara uma função para que o compilador possa identifica-la antes de compilar o programa.

- **Exemplo de um Protótipo de uma Função de multiplicação entre dois números:**

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

```
// Protótipo da função
```

```
int mult(int n1, int n2);
```

```
int main(void)
```

```
{
```

```
int v1, v2, result;
```

```
printf("Digite o primeiro valor:");
```

```
scanf("%d", &v1);
```

```
printf("Digite o segundo valor:");
```

```
scanf("%d", &v2);
```

```
//chama a função e recebe o retorno
```

```
result = mult(v1,v2);
```

```
printf("Resultado = %d\n", result);
```

```
return 0;
```

```
}
```

```
int mult(int n1, int n2)
```

```
//mult recebe n1e n2 e retorna um int
```

```
{
```

```
int result;
```

```
result = n1 * n2;
```

```
return(result);
```

```
//retorna o valor para main()
```

```
}
```

Função como parâmetro de outra Função

Após vermos que uma função pode enviar um valor como retorno para o programa, podemos então fazer com que o valor desse retorno chame uma nova função. Fazendo com que utilizemos uma função para ser o parâmetro (retorno) de outra função.

- **Exemplo de Função como parâmetro de outra Função onde se encontra a multiplicação do quadrado de dois números:**

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

```
int multq(int n1, int n2);
```

```
int qdr(int q);
```

```
int mult(int m1, int m2);
```

```
int main()
```

```
{
```

```

int v1,v2;
printf("Digite o primeiro valor:");
scanf("%d",&v1);
printf("Digite o segundo valor:");
scanf("%d",&v2);
printf("Resultado = %d\n",multq(v1,v2));
return 0;
}
int multq(int n1, int n2)
{
    return mult(qdr(n1),qdr(n2));
}
int qdr(int q)
{
    return q*q;
}
int mult(int m1, int m2)
{
    return m1*m2;
}

```

Função Recursiva

E por fim, temos a função recursiva. A função recursiva é uma função de programação onde o valor retornado chama a própria função. Temos que tomar cuidado para uma função recursiva não entrar em loop, se chamando infinitas vezes. Para isso temos que interromper a função em dado momento.

- **Exemplo de Função Recursiva com o fatorial de um número:**

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

```
int fat(int n)
```

```
{  
    if (n)  
        return n*fat(n-1);  
    else return 1;  
}  
int main()  
{  
    int n;  
    printf("Digite um valor para n: ");  
    scanf("%d", &n);  
    printf("Resultado = %d", n, fat(n));  
    return 0;  
}
```