

# {introcomp}

## Working 05 : Modularização

### Objetivos:

- Entender como é estruturada uma função em C e lembrar como é feita a comunicação entre funções;
- Praticar a modularização em algoritmos um pouco mais complexos;

**Prazo de Envio:** sábado, 23/09, 04:00.

# 1 INTRODUÇÃO

Vimos que quando os programas vão ficando mais complexos e maiores, surge a necessidade de dividi-los em subprogramas menores e juntá-los na função principal (main) para que seja gerada a saída esperada. Essa prática é conhecida como modularização. Em C, a modularização é feita através de funções e subfunções, que se comunicam entre si por meio da entrada (parâmetros) e saída (retorno) de dados. Vale lembrar que não necessariamente uma função deve conter parâmetros ou retorno.

## 2 ESTRUTURA DE UMA FUNÇÃO

Vejam agora o que compõe uma função:

### 2.1 Cabeçalho:

<tipoRetorno> <nome> (<tipo> <nomeParametro1>, ..., <tipo> <nomeParametroN>)

### 2.2 Exemplos:

```
[1])float calculaMedia (float n1, float n2, float n3) int dobraValor (int valor) char funcaoBolada (float real, int inteiro, char letra)
```

Note que uma função pode ter quantos parâmetros o programador achar necessário, porém só pode conter um retorno.

### 2.3 Dicionário de dados:

No dicionário de dados deve-se fazer a declaração das variáveis que serão utilizadas na função. Essas variáveis são chamadas de variáveis locais e só podem ser usadas na função que as criou. Após o término da função (quando houver retorno de um dado ou for lida completamente) as variáveis locais são apagadas da memória do computador.

Lembre-se também que as variáveis da função principal ou de alguma função não são enxergadas por outras funções. Apenas variáveis globais são vistas igualmente por todas. Com isso, pode-se declarar variáveis com o mesmo nome em funções distintas, o que não pode ocorrer com variáveis globais.

### 2.4 Corpo:

O corpo da função é o próprio algoritmo, o processamento da função. É no corpo em que são feitos os loops, comparações, variáveis, retorno da função etc.

### 2.5 Comentários:

É muito importante sempre deixar comentários dizendo o que a função faz e a que cada variável se refere. Além de ajudar outras pessoas a entenderem o seu código, quando for

reaproveitar as funções em outros programas facilitará lembrar o funcionamento dela.

### Exemplo:

```
1 #include <stdio.h>
3 float calculamedia(float n1, float n2){
4     int soma = n1 + n2;
5     int media = soma/2;
6     return media;
7 } //recebe dois numeros e faz a media entre eles.
9 int main(){
10     float nota1, nota2;
11     scanf("%f %f",&nota1,&nota2);
12     float media = calculamedia(nota1, nota2);
13     printf("A media dos valores e igual a: %.2f", media);
14     return 0;
15 }
```

**Obs:** a função é feita fora da main! E só é utilizada pelo programa no momento em que for chamada.

## 2.6 Função sem retorno de dados:

Algumas funções não precisam retornar dados, então pra isso existe o tipo void. Geralmente essas funções tem como objetivo apenas imprimir na tela uma mensagem.

## 2.7 Função sem parâmetros:

Algumas funções podem ter retorno de dados, mas não precisarem de parâmetros. Basta retornar um valor calculado ou adquirido dentro da própria função (se a função ler as informações necessárias do teclado por exemplo então não precisará de parâmetros).

# Praticando

OBS: Os códigos desses praticandos devem ser modularizados da melhor maneira possível. Mesmo nos praticandos em que é pedido apenas para criar uma função, você deve criar a main de forma a ser possível realizar os testes necessários.

1. Crie um programa que receba do teclado dois números reais e um caractere que represente uma operação entre eles (forma de recebimento: <número><caractere><número>) e com uma função, faça a operação desejada. Obs: Possíveis caracteres: +, -, \*, /

Exemplo de Entrada	Exemplo de Saida
5.0 + 7.0	12.0
2.0 * 5.0	10.0

2. Faça um programa que leia uma letra do teclado, verifique se ele é maiúsculo ou minúsculo e imprima na tela. Crie uma função para essa verificação. Obs: será dada somente letra como entrada.

Exemplo de Entrada	Exemplo de Saida
H	A letra e maiuscula
h	A letra e minuscula

3. Faça um programa que leia números digitados pelo teclado até que um deles seja -1. Para cada número digitado, diga se ele é par ou ímpar criando uma função para isto.

Exemplo de Entrada	Exemplo de Saida
10	O numero e par
1	O numero e impar
9	O numero e impar
-1	

4. Faça um programa que leia um número N seguido de N notas (valores reais entre 0 e 10). Mostre no terminal a maior nota, a menor nota e a média da turma.

Exemplo de Entrada	Exemplo de Saida
5 10 7 6.75 2.5 8.5	10.00 2.50 6.95
4 9 9 9 6	9.00 6.00 8.25

5. Faça um programa que leia varios caracteres separados por ponto e virgula (;) e mostre quantos caracteres são maiusculos e quantos são minusculos, respectivamente. O fim da leitura será sinalizado por um ponto (.) .  
Dica: Utilize a função criada no exercicio 2

Exemplo de Entrada(Arquivo)	Exemplo de Saida(Terminal)
a;B;c;d;e;F.	2 4
Z;A;N;H;Y;L.	6 0

## DESAFIOS

1. Faça um programa que dado um número inteiro, informe quantos dígitos ele possui.  
Crie uma função para verificar a quantidade de dígitos.

Exemplo de Entrada	Exemplo de Saida
10567823	8