



Sub-rotinas em Linguagem C

Sub-rotinas: vantagens

- A divisão de um programa em **blocos lógicos** facilita a construção de uma solução (ou a análise de uma solução que já esteja pronta);
- Reaproveitamento de código;
- Facilita a identificação de problemas;
- O uso será *estimulado* no curso!

Exemplo: divisão em **blocos lógicos**.

```
#include <stdio.h>

int main()
{
    int a, b;

    a = LerValorEntre(5, 150);
    b = LerValorEntre(0, 100);
    ExibirMaior(a, b);
    return 0;
}
```

Exemplo: a **definição** de uma função...

```
float Media (int a, int b, int c)  
{  
    float M;  
    M = (a+b+c) / 3.0;  
    return M;  
}
```

a, b, c são
parâmetros e M
variável local.

e a sua **chamada**:

```
m = Media (a, b, 5);
```

Variáveis a e b são
locais da função que
chamou a **Media**

Sub-rotinas em C

Em linguagem C, toda sub-rotina é uma **função** porque **sempre** há **retorno** de valor. Lembrando:

`y` = `sqrt`(`x * x`);

Variável para o
valor de retorno

Nome da função

Argumentos

Evite fazer entrada e exibição de dados em funções que realizam cálculos.

Uma função pode conter **mais de uma** instrução de retorno. Exemplo:

```
...  
if (x >= 0)  
    return x;  
else  
    return -x;  
...
```

*A execução da função termina quando a instrução **return** (qualquer uma delas...) é executada.*

Não entendi...

Epa! Mas não tem retorno de valor em

```
printf("Alô mundo");
```

*A função **printf** retorna valor sim, apesar dele (normalmente) não ser usado.*

E quando eu uso

```
system("CLS");    ?
```

*Esta função retorna um valor do tipo **void** (= vazio, ou seja, a função não retorna valor).*

A **definição** de uma função deve ser escrita **antes** da chamada. Opção: definir um **protótipo** para a função, antes do ponto de chamada. Exemplo:

```
float Media (int, int, int) ;
```

***Estratégia:** definir globalmente os **protótipos** das funções, **chamá-las** no programa principal (ou dentro de outras funções) e escrever as **definições** depois do código do programa principal (ou em outro arquivo).*

Protótipos: legibilidade!

```
#include <stdio.h>

int LerValorEntre(int,int) ; /* Protótipo */
void ExibirMaior(int,int) ; /* Protótipo */

main()
{
    int a,b;

    a = LerValorEntre(5,150);
    b = LerValorEntre(0,100);
    ExibirMaior(a,b);
}
```

Vamos
testar!



*Eu consigo entender o **quê** o programa faz, mesmo sem ter certeza de **como** as funções devem ser definidas...*