

Subprogramas

Subprogramas são trechos de programa que realizam uma tarefa específica. Podem ser chamados pelo nome a partir do programa principal ou de trechos de outros subprogramas, até mesmo ele próprio (chamada recursiva).

Podem ser de dois tipos:

Funções e Procedimentos

A linguagem C possui apenas o Subprograma do tipo função.

Porque usar funções ?

- Para permitir o reaproveitamento de código já construído(por você ou por outros programadores);
- Para evitar que um trecho de código que seja repetido várias vezes dentro de um mesmo programa;
- Para permitir a alteração de um trecho de código de uma forma mais rápida. Com o uso de uma função é preciso alterar apenas **dentro** da função que se deseja;
- Para que os blocos do programa não fiquem grandes demais e, por consequência, mais difíceis de entender;
- Para facilitar a leitura do programa-fonte de uma forma mais fácil;
- Para separar o programa em partes(blocos) que possam ser logicamente compreendidos de forma isolada.

Formato Geral de uma Função em C

```
tipo_da_funcao NomeDaFuncao (Lista_de_Parâmetros)
{
// corpo da função
}
```

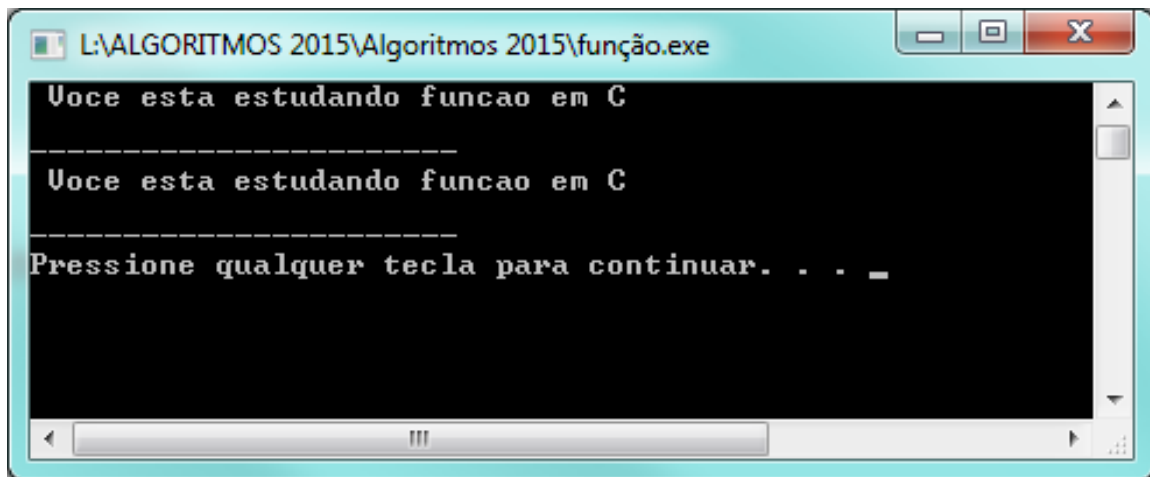
OBS * Lista de Parâmetros pode ser vazia**

Função sem Parâmetros

```
void printa( )
{
printf(" Voce esta estudando funcao em C  \n");
printf("\n");
}

void traco()
{
printf("----- \n");
}

{
printa();
traco();
printa();
traco();
system("pause");
}
```



Observe que neste caso a função faz sempre a mesma coisa

Função com Parâmetros

Ex: Cabeçalho da função

float media (float nota1, float nota2)

Parâmetros formais

Chamada:

< nome_funcao > (< parametros_reais >)

Parâmetros reais

Ex: media(8,10);

Para que a função retorne um valor em seu nome, é usada a instrução **return**.

Ex:

```
float_media(float nota1, float nota2)
{
    float x;
    x=(nota1+nota2)/2;
    return(x);
}
```

x é o resultado do cálculo

Parâmetros formais - são aqueles passados na declaração da função. É onde informamos quais as variáveis que a função irá receber quando chamada e quais os seus tipos (são informados como uma declaração das variáveis). Esses parâmetros são considerados como variáveis locais a função. Se a função não precisa receber nenhum parametro, colocamos entre os parênteses a palavra **void**.

Parâmetros reais - são aqueles passados na chamada da função. É quando informamos quais os valores que os parâmetros terão dentro da função. Se a função não espera receber nenhum parâmetro, na sua chamada colocamos o abre e fecha parênteses vazio

****Programa principal (main) e a função**

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

```
float media(float nota1, float nota2)
{
    float x;
    x=(nota1+nota2)/2;
    return x;
}
```

```
main()
{
    float a,b,result;

    printf("entre com dois valores  \n");
    scanf("%f",&a);
    scanf("%f",&b);
    result=media(a,b);
    printf("a media e =  %.2f",result);
    getch();
}
```

Passagem de Parâmetros

Os parâmetros formais (variáveis locais a função chamada) são inicializados com o valor dos parâmetros reais.

Passagem por valor - O valor dos parâmetros formais sendo alterado durante a execução da função não acarretará em nenhuma modificação no valor dos parâmetros reais (variáveis da função chamadora).

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

// passagem por valor

void troca(int a,int b)
{
    int aux;
    aux=a;
    a=b;
    b=aux;
    printf(" dentro da funcao  %d %d\n",a,b);
}

main()
{
    int x=5;
    int y=10;

    printf("%d %d\n    ",x,y);
    troca (x,y);
    printf("%d %d    ",x,y);
    getch();
}
```

Passagem por referência - Qualquer alteração no valor dos parâmetros formais durante a execução da função será refletida no valor de seus parâmetros reais correspondentes.

A linguagem C não implementa automaticamente a passagem de parâmetros por referência. Esta deve ser simulada através da utilização de ponteiros. Ao invés de passarmos a variável como parâmetro, passamos o endereço desta variável. Sendo assim, a função que recebe pode utilizar seu valor através de indireção se ele for alterado essa alteração irá refletir fora da função.

Ex:

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

void troca(int &a,int &b);

main()
{
int x=5;
int y=10;
printf("%d %d\n",x,y);
troca (x,y);
printf("%d %d",x,y);
}

void troca(int &a,int &b)
{
int aux;
aux=a;
a=b;
b=aux;
}
```