



SCC120 - Capítulo 5

Funções em C

João Luís Garcia Rosa

Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - São Carlos
<http://www.icmc.usp.br/~joaoluis>
2010

1

Operadores & e *

```
#include <stdio.h>

main()
{
    int x = 15;
    printf("Endereco da variavel x = %d.\n", &x);
    printf("Conteudo do endereco da variavel x =
    %d.\n", *(&x));
}
```

Saída:

Endereco da variavel x = 1245052.

Conteudo do endereco da variavel x = 15.

2

Ponteiros e const

```
int gorp = 16;  
int chips = 12;  
const int * p_snack = &gorp;
```

```
*p_snack = 20; // NÃO
```

não permite a mudança de valor para o qual *p_snack* aponta

```
p_snack = &chips; // OK
```

p_snack pode apontar para uma outra variável

```
int gorp = 16;  
int chips = 12;  
int * const p_snack = &gorp;
```

```
*p_snack = 20; // OK
```

p_snack pode ser usado para mudar valor

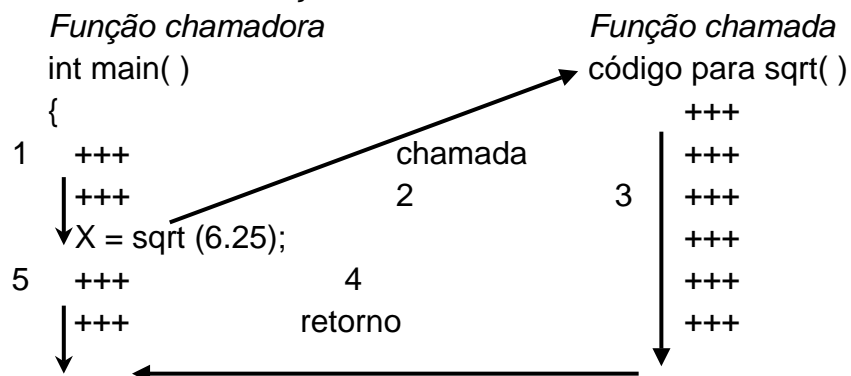
```
p_snack = &chips; // NÃO
```

não permite a mudança de variável para o qual *p_snack* aponta

3

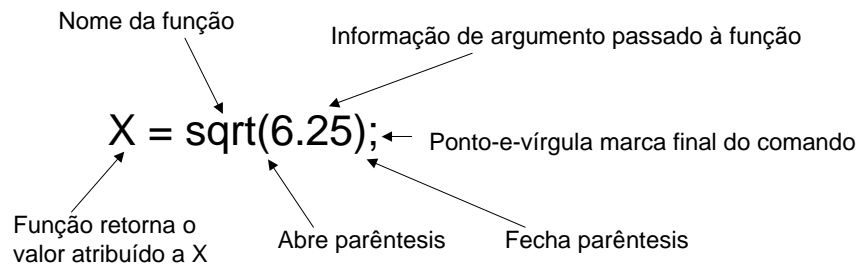
Usando uma função com um valor de retorno

Chamando uma função:



4

Sintaxe da chamada da função:



5

```
#include <stdio.h>
#include <math.h>
main()
{
    float cobertura; // usa float para números reais
    float lado;
    printf("Quantos centímetros quadrados de folhas
voce tem?\n");
    scanf("%g", &cobertura);
    lado = (float) sqrt(cobertura); // chama a função,
// atribuindo valor de retorno
    printf("Voce pode cobrir um quadrado com lados de
%g", lado);
    printf(" centímetros\ncom suas folhas.\n");
}
```

6

Saída:

Quantos centímetros quadrados de folhas você tem?

123.21

Você pode cobrir um quadrado com lados de 11.1 centímetros

com suas folhas.

7

Funções definidas pelo usuário

```
#include <stdio.h>
void simples(); // protótipo da função

main()
{
    printf("main() chamara a funcao simples():\n");
    simples(); // chamada da função
}

// definição da função
void simples()
{
    printf("Eu sou apenas uma funcao simples.\n");
}
```

8

```
#include <stdio.h>
void simon(int); // protótipo de função para simon()
void main()
{
    int cont;
    simon(3); // chama a função simon()
    printf("Pegue um inteiro: ");
    scanf("%d", &cont);
    simon(cont); // chama-a de novo
}
void simon(int n) // define a função simon()
{
    printf("Simon diz: toque seus dedos do pe %d
vezes.\n", n);
}
// funções void não precisam de comando de retorno 9
```

Saída:

Simon diz: toque seus dedos do pé 3 vezes.

Pegue um inteiro: 512

Simon diz: toque seus dedos do pé 512 vezes.

protótipos de função	<code>#include <iostream.h></code>
	<code>void simon(int);</code>
	<code>double taxas(double);</code>
	<code>void main()</code>
	<code>{</code>
função # 1	<code>...</code>
	<code>}</code>
	<code>void simon(int n)</code>
	<code>{</code>
função # 2	<code>...</code>
	<code>}</code>
	<code>double taxas (double t)</code>
	<code>{</code>
função # 3	<code>...</code>
	<code>return (2 * t);</code>
	<code>}</code>

Definições de funções ocorrem seqüencialmente em um arquivo

11

Função definida pelo usuário com um valor de retorno

```
#include <stdio.h>
int kilopgrama(int); // protótipo da função
main()
{
    int kilo, grama;
    printf("Entre com o peso em Quilogramas: ");
    scanf("%d", &kilo);
    grama = kilopgrama(kilo);
    printf("%d Quilogramas sao ", kilo);
    printf("%d gramas.\n", grama);
}
int kilopgrama(int sts)
{
    return 1000 * sts;
}
```

12

Protótipo de Função e chamadas de função

```
#include <stdio.h>
void saude(int); // protótipo: não retorna valor
float cubo(float x); // protótipo: retorna um double
void main(void)
{
    float lado, volume;
    saude(5); // chamada da função
    printf("De-me um numero: ");
    scanf("%g", &lado);
    volume = cubo(lado); // chamada da função
    printf("Um cubo de %g cm tem um volume de ",
        lado);
    printf("%g cm cubicos.\n", volume);
    saude((int) cubo(2));
}
```

13

```
void saude(int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("Saude! ");
    printf("\n");
}
float cubo(float x)
{
    return x * x * x;
}
```

Saída:

Saúde! Saúde! Saúde! Saúde! Saúde!

Dê-me um número: 5

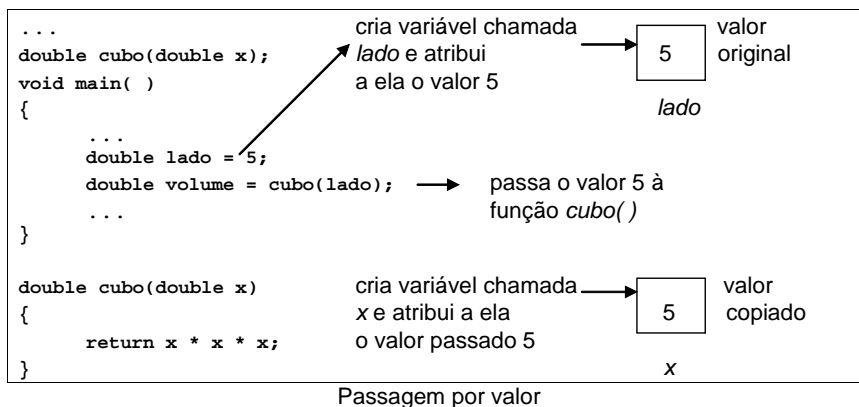
Um cubo de 5 cm tem um volume de 125 cm cúbicos.

Saúde! Saúde! Saúde! Saúde! Saúde! Saúde! Saúde! Saúde!
Saúde!

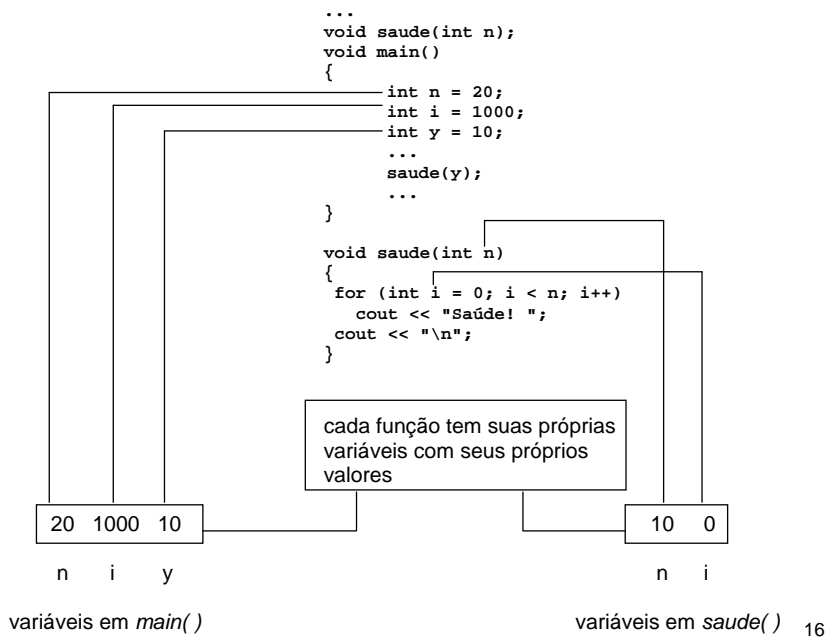
14

Argumentos de Função e Passagem por Valor

Argumentos de Função e Passagem por Valor



Variáveis Locais:




```

#include <stdio.h>
#include <conio.h>
void n_chars(char, int);
void main()
{
    int vezes = 0;
    char ch;
    printf("Entre com um caractere: ");
    ch = getche();
    while (ch != 's') // s para sair
    {
        printf("\nEntre com um inteiro: ");
        scanf("%d", &vezes);
        n_chars(ch, vezes); // função com dois argumentos
        printf("\nEntre com um outro caractere ou pressione a
tecla s para sair: ");
        ch = getche();
    }
    printf("\nO valor de vezes eh %d.\n", vezes);
    printf("Tchau\n");
}
void n_chars(char c, int n) // mostra c n vezes
{
    while (n-- > 0) // continua até que n alcance 0
        printf("%c", c);
}

```

17

Argumentos Múltiplos

Saída:

```

Entre com um caractere: W
Entre com um inteiro: 50
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Entre com um outro caractere ou pressione a tecla s
para sair: a
Entre com um inteiro: 20
aaaaaaaaaaaaaaaaaaaaaaaaa
Entre com um outro caractere ou pressione a tecla s
para sair: s
O valor de vezes é 20.
Tchau

```

18

```

#include <stdio.h>
float chance(unsigned numeros, unsigned pegadas);
void main()
{
    float total, escolhas;
    printf("Entre com o numero total de escolhas de
    cartoes de jogo e \n");
    printf("numero de escolhas pegadas permitida:\n");
    while (scanf("%f%f", &total, &escolhas) &&
    escolhas <= total)
    {
        printf("Voce tem uma chance em ");
        printf("%g", chance((int)total,
        (int)escolhas)); // computa a chance
        printf(" de ganhar.\n");
        printf("Proximos dois numeros (s para sair:
        ");
    }
    printf("tchau\n");
}

```

19

Uma Outra função de dois argumentos

```

// a seguinte função calcula a chance de pegar os
// números corretamente a partir de escolhas de
// números

float chance(unsigned numeros, unsigned pegadas)
{
    float result = 1.0; // algumas variáveis locais
    unsigned n;
    unsigned p;
    for (n = numeros, p = pegadas; p > 0; n--, p--)
        result = result * n / p ;
    return result;
}

```

20

Saída:

Entre com o número total de escolhas de cartões de jogo e

número de escolhas pegadas permitida:

49 6

Você tem uma chance em 1.39838e+07 de ganhar.

Próximos dois números (s para sair): 51 6

Você tem uma chance em 1.80095e+07 de ganhar.

Próximos dois números (s para sair): 38 6

Você tem uma chance em 2.76068e+06 de ganhar.

Próximos dois números (s para sair): s

tchau

21

Funções e Strings no Estilo C

```
#include <stdio.h>

int c_in_str(const char * str, char ch);

void main()
{
    char mmm[15] = "minimum"; // string em um vetor
    char *uivo = "ululado"; // uivo aponta para string
    int ms = c_in_str(mmm, 'm');
    int us = c_in_str(uivo, 'u');
    printf("%d caracteres m em %s\n", ms, mmm);
    printf("%d caracteres u em %s\n", us, uivo);
}
```

22

```
// esta função conta o número de caracteres ch
// na string str
```

```
int c_in_str(const char * str, char ch)
{
    int cont = 0;
    while (*str) // sai quando *str é '\0'
    {
        if (*str == ch)
            cont++;
        str++; // move ponteiro para próximo char
    }
    return cont;
}
```

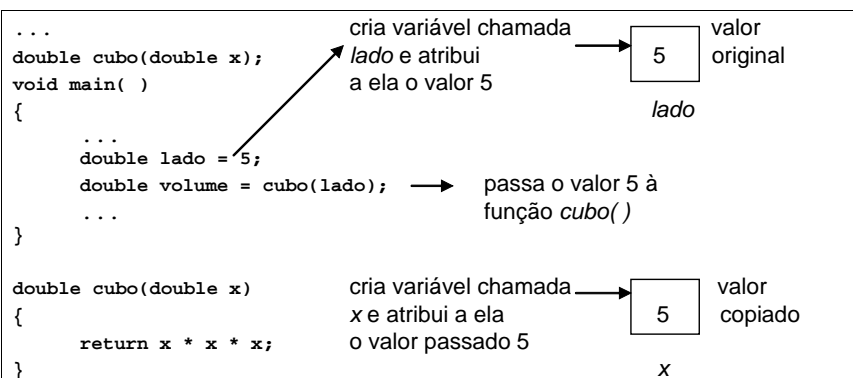
Saída:

```
3 caracteres m em minimum
2 caracteres u em ululado
```

23

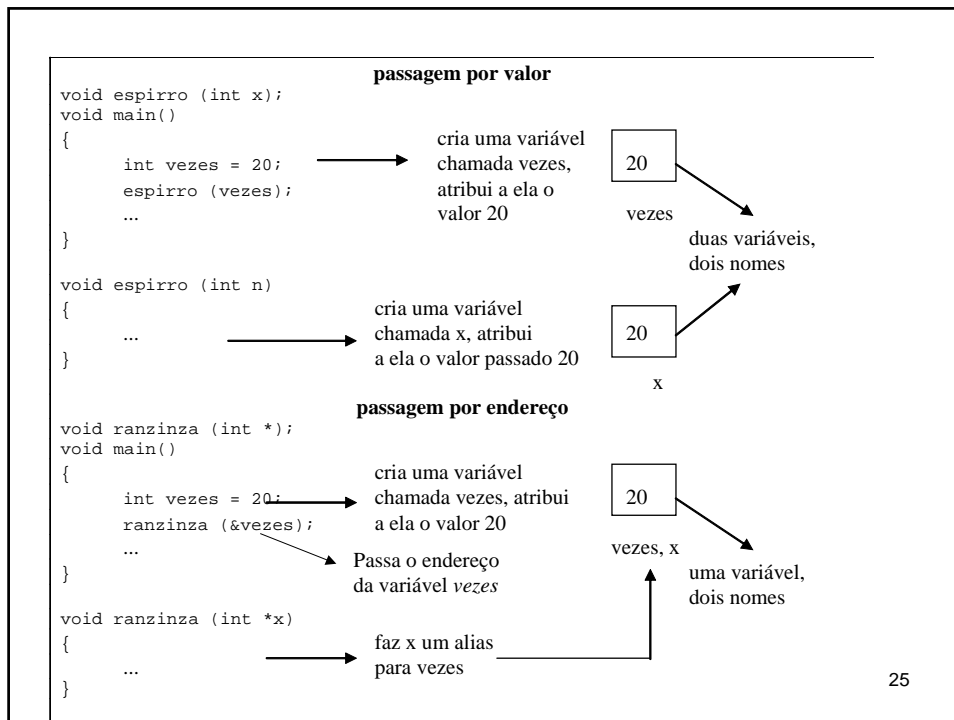
Argumentos de Função e Passagens por Valor e Endereço

Argumentos de Função e Passagem por Valor



Passagem por valor

24



```

#include <stdio.h>
altera(int *, int *);
main()
{
    int x,y;
    printf("Entre com o primeiro valor: ");
    scanf("%d", &x);
    printf("Entre com o segundo valor: ");
    scanf("%d", &y);
    altera(&x,&y);
}

altera(int *px,int *py)
{
    int soma, dif;
    soma = *px + *py;
    dif = *px - *py;
    *px = soma;
    *py = dif;
    printf("Soma = %d ou %d; Diferença = %d ou %d\n", soma,
    *px, 135, dif, *py);
}

```

26

Lê um caractere por vez

```
#include <stdio.h>
main()
{
    char ch;
    int count = 0; // usa entrada básica
    scanf("%c", &ch); // pega um caractere
    while (ch != '#') // testa o caractere
    {
        if (ch != ' ')
        {
            printf("%c", ch); // ecoa o caractere
            count++; // conta o caractere
        }
        scanf ("%c", &ch); // pega o próximo caractere
    }
    printf("\n%d caracteres lidos\n", count);
}
```

27

Lê a string inteira

```
#include <stdio.h>
main()
{
    char ch[50];
    int i = 0;
    int count = 0; // usa entrada básica
    gets(ch); // pega uma string
    while (ch[i] != '#' && ch[i] != '\0') // testa
    {
        if (ch[i] != ' ')
        {
            printf("%c", ch[i]); // ecoa o caractere
            count++; // conta o caractere ecoado
        }
        i++; // próximo caractere
    }
    printf("\n%d caracteres ecoados\n", count);
}
```

28

Funções que retornam strings

```
#include <stdio.h>
char * conststr(char c, int n); // protótipo
char str[20];
main()
{
    int vezes;
    char ch;
    char * ps = str;
    printf("Entre com um caractere: ");
    scanf("%c", &ch);
    printf("Entre com um inteiro: ");
    scanf("%d", &vezes);
    ps = conststr(ch, vezes);
    printf("%s\n", ps);
    ps = conststr('+', 20); // reusa ponteiro
    printf("%d-FIM-%s\n", ps, ps);
}
```

29

```
// contrói string feita de n caracteres c
char * conststr(char c, int n)
{
    char * pstr = str;
    pstr[n] = '\0'; // termina string
    while (n-- > 0)
        pstr[n] = c; // preenche resto da string
    return pstr;
}
```

Saída:

```
Entre com um caractere: R
Entre com um inteiro: 20
RRRRRRRRRRRRRRRRRRRRRR
4347904-FIM-+++++
```

30

EXERCÍCIOS PROPOSTOS

1. Escreva uma função que retorne a soma de dois inteiros.
2. O que há de errado com a definição desta função:

```
salame(num)
{
    int num, cont;
    for(cont = 1; cont <= num; num++)
        printf(" Ó salame mio!\n");
}
```

3. Escreva uma função `max(n,m)` que retorne o maior de dois valores.

31

4. Use *loops* aninhados para escrever um programa que produza este padrão:

```
$$$$$$$$$
$$$$$$$$$
$$$$$$$$$
$$$$$$$$$
```

5. Que saída produzirá o seguinte *loop*?

```
for ( valor = 36; valor > 0; valor /= 2)
    printf("%3d", valor);
```

32

Respostas

```
// EP01
#include <stdio.h>
int soma(int, int);

main()
{
    int x,y;

    printf("Entre com o primeiro inteiro: ");
    scanf("%d", &x);
    printf("Entre com o segundo inteiro: ");
    scanf("%d", &y);
    printf("A soma de %d e %d %c %d\n", x, y, 130, soma(x,y));
}

int soma(int a, int b)
{
    return a + b;
}
```

33

```
// EP02
#include <stdio.h>

salame(int);

main()
{
    salame(5);
}

salame(int num)
{
    int cont;
    for(cont = 1; cont <= num; cont++)
        printf(" %c salame mio!\n", 224);
}
```

34

```

// EP03
#include <stdio.h>
int max(int, int);

main()
{
    int x,y;

    printf("Entre com o primeiro inteiro: ");
    scanf("%d", &x);
    printf("Entre com o segundo inteiro: ");
    scanf("%d", &y);
    printf("O maior entre %d e %d %c %d\n", x, y, 130, max(x,y));
}

int max(int a, int b)
{
    return a > b ? a : b; // retorna a se a > b, senão retorna b
}

```

35

```

// EP04
#include <stdio.h>

main()
{
    int i,j;

    for (i = 1; i <= 4; i++)
    {
        for (j = 1; j <= 8; j++)
            printf("$");
        printf("\n");
    }
}

```

36

```
// EP05:
```

```
36 18 9 4 2 1
```

37

```
#include <stdio.h>
alter(int *, int *);
main()
{
    int x,y;
    printf("Entre com o primeiro valor: ");
    scanf("%d", &x);
    printf("Entre com o segundo valor: ");
    scanf("%d", &y);
    alter(&x,&y);
    printf("\nNovos valores: x = %d e y = %d\n", x, y);
}
alter(int *px,int *py)
{
    int soma, dif;
    soma = *px + *py;
    dif = *px - *py;
    *px = soma;
    *py = dif;
    printf("Soma = %d ou %d; Diferen%ca = %d ou %d\n", soma, *px, 135, dif,
    *py);
}
```

Saída:

```
Entre com o primeiro valor: 3
Entre com o segundo valor: 7
Soma = 10 ou 10; Diferença = -4 ou -4
```

```
Novos valores: x = 10 e y = -4
```

38

```

#include <stdio.h>
main()
{
    char ch[50];
    char * c = ch;    // string c recebe espaço da string ch
    int i = 0;
    int count = 0;    // usa entrada básica
    gets(c);         // pega uma string
    while (*c != '#' && *c) // testa o caractere
    {
        if (*c != ' ')
        {
            printf("%c", *c);    // ecoa o caractere
            count++;             // conta o caractere ecoado
        }
        c++;                     // próximo endereço
    }
    printf("\n%d caracteres ecoados\n", count);
}

```

39

Referência

- Prata, S. *C++ Primer Plus*. Waite Group Press, 1998.