INSTITUTO FEDERAL
RIO GRANDE DO SUL

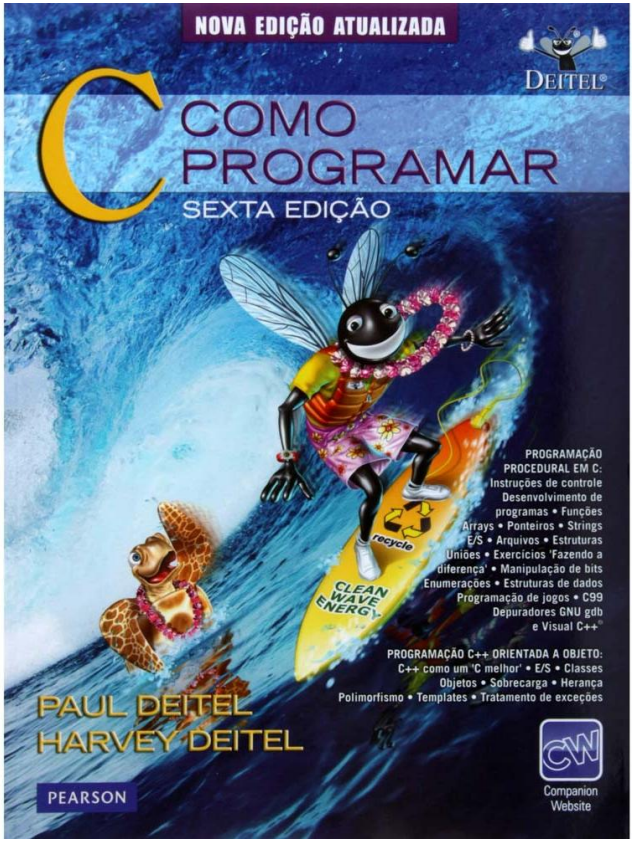
# Algoritmos e Programação II

# Procedimentos e Funções

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Bibliografia




**Titulo**: Como Programa C – Sexta Edição
**Editora:** Pearson
**Autor**: Paul Deitel

**Capítulo 5.** Página 186

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Procedimentos e Funções

- A maioria dos programas (codificação) de computador que resolvem problemas do mundo real são gigantescos (milhares de linhas de código).
- A experiência tem mostrado que a melhor maneira de desenvolver e manter um programa grande é construí-lo a partir de pequenas partes ou *módulos, sendo cada uma delas mais fácil de* manipular do que o programa original.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Procedimentos e Funções

- Os Módulos em C são chamados de procedimentos ou funções. Geralmente, os programas em C são escritos combinando novas funções que o programador escreve com funções "prontas“ disponíveis na *biblioteca padrão do C (C standart library).*

- A biblioteca padrão do C fornece um excelente conjunto de funções para realizar cálculos matemáticos comuns, manipulação de strings, manipulação de caracteres, entrada/saída e muitas outras operações.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Procedimentos e Funções

- O programador pode escrever procedimentos ou funções para definir tarefas específicas e que podem ser utilizadas em muitos locais dos programas.

- Também pode ser utilizado para trechos do programada que são chamadas ou invocadas em vários partes do código.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Procedimentos e Funções

- Cada função deve se limitar a realizar uma tarefa simples e bem definida, e o nome da função deve expressar efetivamente aquela tarefa. Isso facilita a abstração e favorece a capacidade de reutilização do software.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Procedimentos vs Funções

| Procedimentos | Funções |
|---|---|
| Procedimentos não retornam um valor. | As funções são avaliadas e devem retornam um valor ao trecho do programa que as invocou (retorna um valor a uma determinada variável) |
| Não possuem obrigatoriedade de parâmetros. | Possuem parâmetros para execução. |
| Um procedimento é ativado através de um comando de chamada do procedimento. | Uma função é ativada quando é avaliada uma expressão que a contém, isto é, as funções são utilizadas da mesma forma que as funções predefinidas, como *printf*, *scanf*, *gets, etc.* |

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Escopo das Variáveis

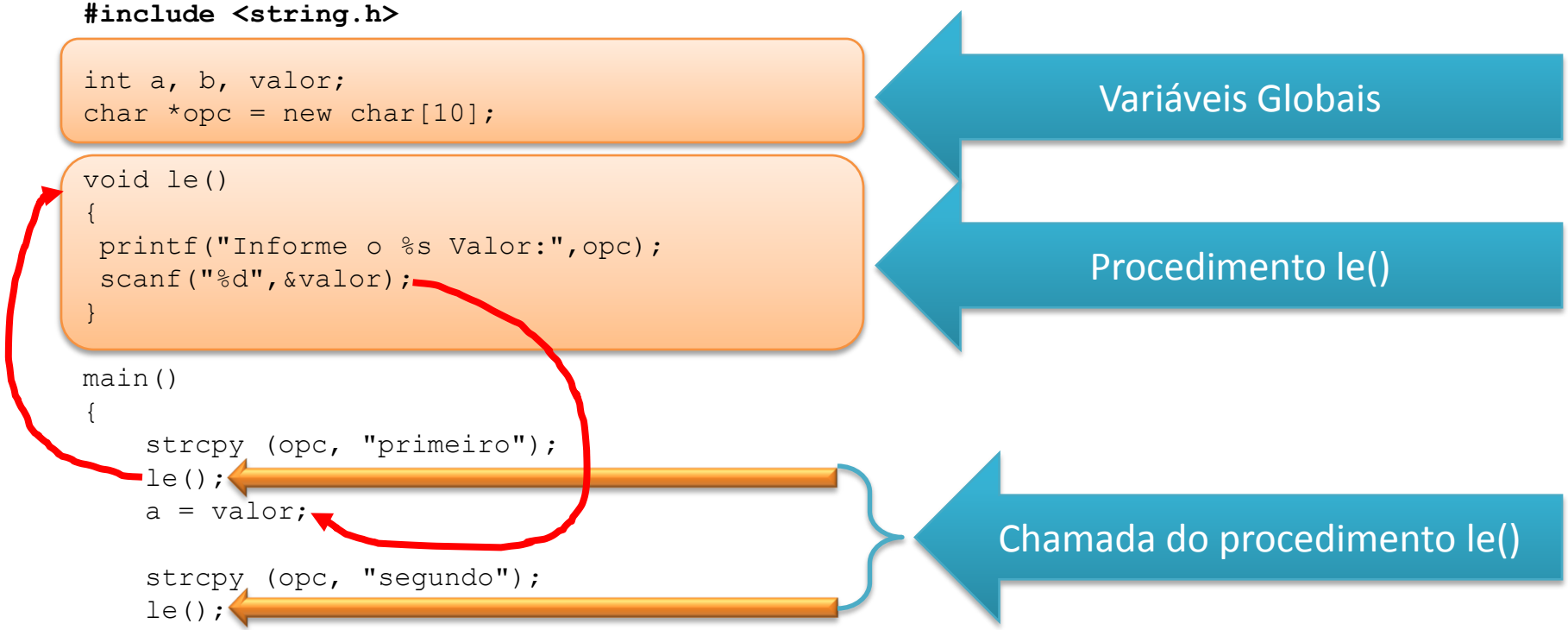
- **Variáveis Globais:** São as variáveis declaradas no programa que são conhecidas em todo programa e inclusive nos subprogramas contidos nele.
- **Variáveis Locais:** São as variáveis declaradas em um subprograma, que são conhecidas em todo o subprograma, mas não no programa que contém o subprograma.
- **Mesmos nomes**: se um subprograma definir nomes de variáveis iguais ao do programa principal, ao referenciar uma variável vale a do escopo local.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Procedimentos

# Procedimentos

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

int a, b, valor;
char *opc = new char[10];

void le()
{
 printf("Informe o %s Valor:",opc);
 scanf("%d",&valor);
}

main()
{
    strcpy (opc, "primeiro");
    le();
    a = valor;

    strcpy (opc, "segundo");
    le();
    b = valor;

    printf("\nVoce digitou os valores %d %d\n", a, b);
    system("pause") ;
}
```



# Procedimentos

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

int a, b, valor;
char *opc = new char[10];

void le();

main()
{

strcpy (opc, "primeiro");
le();
a = valor;

strcpy (opc, "segundo");
le();
b = valor;

printf("\nVoce digitou os valores %d %d\n", a, b);
system("pause") ;
}

void le()
{
 printf("Informe o %s Valor:",opc);
 scanf("%d",&valor);
}
```

Protótipo do procedimento (definição)

Corpo do procedimento

# Procedimentos

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int a, b, valor;
char *opc = new char[10];

void le();
void limpa();

main()
{       strcpy (opc, "primeiro");
        le();
        a = valor;
        strcpy (opc, "segundo");
        le();
        b = valor;
        limpa();
        printf("\nVoce digitou os valores %d %d\n", a, b);
        system("pause") ;
}
```

```
void le()
{
    limpa();
    printf("Informe o %s Valor:",opc);
    scanf("%d",&valor);
}
```

```
void limpa()
{
    system("cls");
}
```

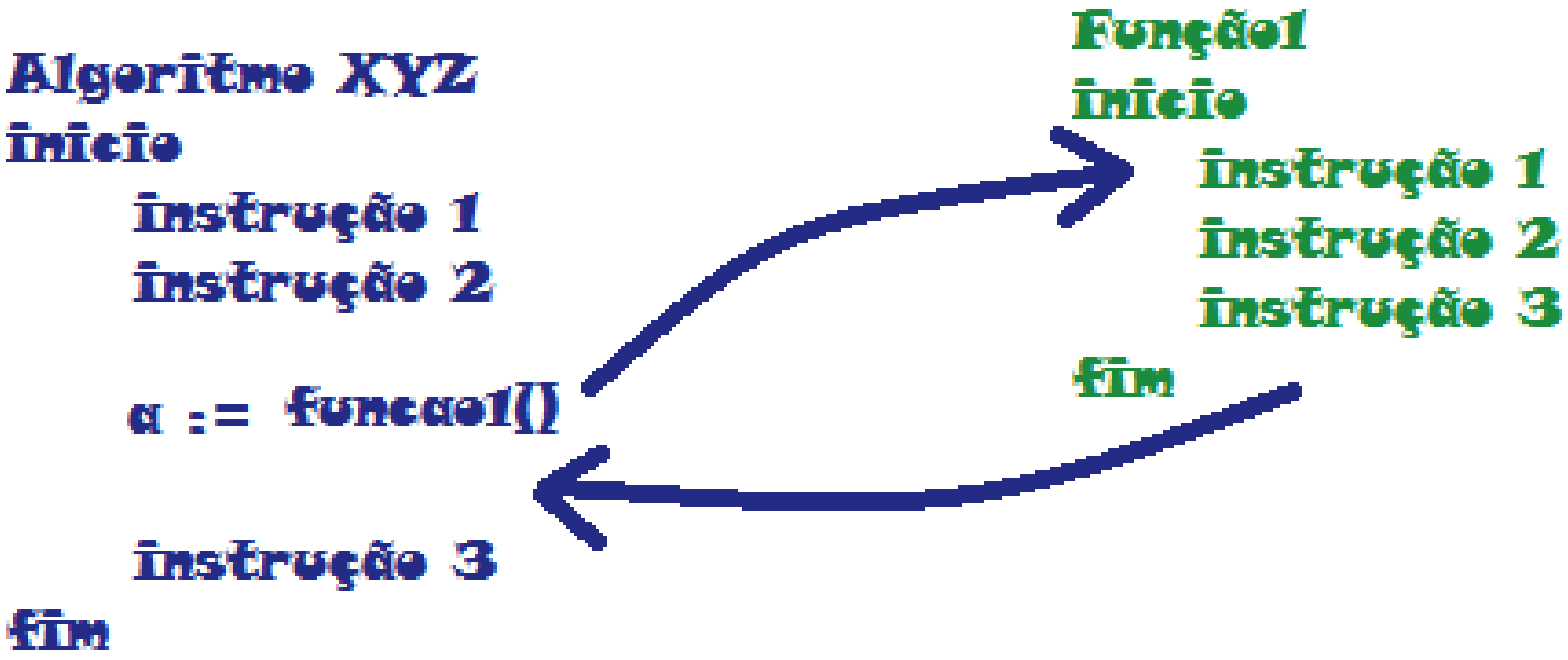
INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Atividade: Procedimentos

- Alterar o código existente, criando um procedimento novo para imprimir os valores inseridos.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Funções



# Funções

Tipo de retorno

Parâmetro

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

int le(char opc[10]);

void limpa();

main()
{
	int a, b;
	a = le("Primeiro");
	b = le("Segundo");
	limpa();
	printf("\nVoce digitou os valores %d %d\n", a, b);
	system("pause") ;
}

int le(char  opc[10])
{
	int valor;
	limpa();
	printf("Informe o %s Valor:",opc);
	scanf("%d",&valor);
	return(valor);


void limpa()
{	system("cls");
}
```

Retorno da função

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Funções

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void limpa()
{
       system("cls");
}
int le(char  opc[10])
{
        int valor;
        limpa();
        printf("Informe o %s Valor:",opc);
        scanf("%d",&valor);
        return(valor);
}
int adicao(int a, int b)
{
       return (a + b);
}
int multiplicacao (int a , int b)
{
       return (a * b);
}
main()
{
      int a, b, soma;
      a = le("Primeiro");
      b = le("Segundo");
      limpa();
      printf("\nVoce digitou os valores %d %d\n", a, b);
      soma = adicao(a, b);
      printf("\nA soma dos valores %d\n", soma);
      printf("\nA multiplicacao dos valores %d\n", multiplicacao (a, b));
      system("pause") ;
}
```


# Funções da Linguagem C

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Funções da Linguagem C

| Função | Descrição | Exemplo |
|---|---|---|
| sqrt(x) | Raiz quadrada de x | sqrt(900.0) é 30<br>sqrt(9.0) é 10 |
| exp(x) | Função exponecial de e^x | exp(1.0) é 2.718282<br>exp(2.0) é 7.389056 |
| log(x) | Logaritmo natural de x (base e) | log (2.718282) é 1.0<br>log (7.389056) é 2.0 |
| log10(x) | Log de x (base 10) | log10(1.0) é 0.0<br>log10(10.0) é 1.0<br>log10(100.0) é 10.0 |
| fabs(x) | Valor absoluto de x | Se x > 0 então fabs(x) é x<br>Se x = 0 então fabs(x) é 0.0<br>Se x < 0 então fabs(x) é –x |
| ceil(x) | Arredonda x para o menor inteiro maior que x | ceil(9.2) é 10<br>ceil(-9.8) é -9 |
| floor (x) | Arredonda x para o maior inteiro menor que x | floor(9.2) é 9<br>floor(-9.8) é -10 |
| pow(x,y) | x elevado a potencia y | pow(2,7) é 128.0<br>pow(9, .5) é 3.0 |
| fmod(x,y) | Resto de x/y, como numero de ponto flutuante | fmod(13.657, 2.333_ é 1.992 |
| sin(x) | Seno trigonométrico de x em rad | sin(0.0) é 0 |
| cos(x) | Cosseno trigonométrico de x em rad | tos(0.0) é 1 |
| tan(x) | Tangente trigonométrico de x em rad | tan(0.0) é 0 |

**Fig. 5.2** Funções da biblioteca matemática usadas normalmente.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Função SRAND()

A função rand() gera uma sequência aleatória de valores.

**void srand (unsigned int x);**

Define uma "semente" para a função rand. Ela deve ser chamada antes do primeiro uso de rand para que a seqüência de números devolvidos por rand não seja sempre a mesma.

**int rand (void);**

Devolve um número inteiro (pseudo)aleatório entre 0 e RAND_MAX inclusive, (#define RAND_MAX 32767), ou seja, um número inteiro no intervalo fechado [0,RAND_MAX].

Se se pretender uma sequência diferente, sempre que o programa é executado, e o utilizador não seja obrigado a introduzir a semente, podemos usar uma função que retorna o valor do relógio do computador em segundos (e cujo protótipo está em time.h):

**int i = srand(time(NULL));**

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Exercícios

1) Crie uma função em C que recebe dois valores (parâmetros) e retorne o maior deles.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Exercícios

2) Crie um procedimento em C que calcule a média de três números digitados e mostre na tela.

O procedimento deverá ser chamado enquanto os três números informados forem diferentes.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Exercícios

3) Fazer um programa para gerar 6 números aleatórios, entre 01 e 60.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

# Exercícios

4) Fazer um programa para popular uma matriz 10x10 com valores aleatórios entre 01 e 100.

O programa deverá ter um **procedimento** para preencher e outro para mostrar a matriz e uma **função** para somar e outra para localizar o maior valor.