Programação Estruturada

Prof. Rodrigo Hausen
http://progest.compscinet.org

Expressões e Funções

VAMOS USAR O LINUX. SE TIVER ENTRADO NO WINDOWS, REINICIE O COMPUTADOR.

TIPOS PRIMITIVOS EM C

Numéricos:

- int
- unsigned int
- long == long int
- unsigned long
- long long
- unsigned long long
- float
- double
- long double
- char (1 byte)
- unsigned char

Textuais:

- char
- unsigned char
- ~~string~~
  array de char

EXPRESSÕES ARITMÉTICAS

Fórmulas matemáticas envolvendo constantes, variáveis e operadores aritméticos.

precedência

maior

	( )		parênteses
	++	--	incremento, decremento
	*	/ %	mult., divisão, resto
⋁	+	-	adição, subtração

menor

EXPRESSÕES ARITMÉTICAS

Qual é o tipo e o resultado das seguintes expressões? Faça de cabeça primeiro, depois verifique no **cling**

- 8 + 5 * 2
- (8 + 5) * 2
- 2015 / 4
- 2015 / 4.0
- 2015 % 4

Faça **int x** = **2**; e calcule $2x^2 + 3x + 1$ (não existe operador para potenciação)

OPERADORES DE INCREMENTO E DECREMENTO

Os operadores ++ e -- fazem o incremento e o decremento de uma variável numérica.

Eles podem ser colocados **antes** ou **depois** de uma variável.

A posição do operador indica em que instante a variável será incrementada: antes ou após a avaliação da expressão.

OPERADORES DE INCREMENTO E DECREMENTO

Execute no **cling**:

```
int i = 0;
i    // imprime 0, valor atual de i
i++  // imprime 0, valor de i antes
     //              de incrementar
i    // imprime 1, valor atual de i
i++  // imprime 1, valor de i antes
     //              de incrementar
i    // imprime 2, valor atual de i
++i  // imprime 3, incrementou antes
i    // imprime 3, valor atual de i
++i  // imprime...
```

OPERADORES DE INCREMENTO E DECREMENTO

Seja i = 2 antes da execução das expressões abaixo, e suponha que elas são executadas na sequência.

Qual é o valor de cada uma e qual o valor de i ao final da execução de ambas? Pense primeiro, depois teste no **cling**

```
5 + (++i)
3 + (i++)
```

OPERADORES DE INCREMENTO E DECREMENTO

Qual é o resultado da execução do código abaixo?

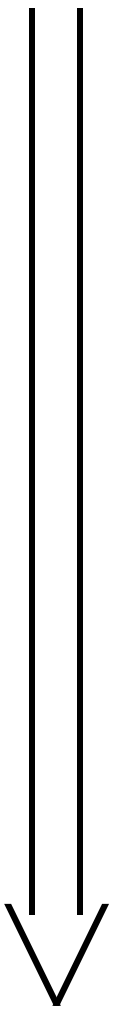
```
int i = 1, j = 2;
2*i*(i++) - (--j)*j
```

Para que complicar, se você pode simplificar? Não seja malvado!

```
--j;
2*i*i - j*j
i++;
```

EXPRESSÕES LÓGICAS

precedência
maior



( )	parênteses
!	negação
< <= > >=	comparação tipos numér.
== !=	igual a, diferente de
&&	**e** lógico
||	**ou** lógico

menor

EXPRESSÕES LÓGICAS

Relembrando: não há tipo primitivo em C para valores booleanos.

Exs.: 1 || 0 resulta 1 (V **ou** F é V)
-1 || 0 resulta 1 (V **ou** F é V

Exer. 1: seja i um int. Escreva uma expressão lógica que é verdadeira se i é ímpar.

Exer. 2: seja ano um int. Escreva uma expressão lógica para determinar se ele é bissexto.

ATRIBUIÇÃO

Em uma expressão, a atribuição é a última operação a ser feita, a menos que esteja entre parênteses.

```
int ano = 2015;
int bissexto;
bissexto = (ano % 400 == 0) ||
  (ano % 100 != 0 && ano % 4 == 0)

int a = 0, b;
b = (a + 1) + (a = 2) + a → res. 5
b = (a + 1) + a = 2 + a → erro sintaxe
```

EXPRESSÃO CONDICIONAL

- A linguagem C possui sintaxe especial para cálculo condicional de expressões

  exprTeste ? *exprV* : *exprF*

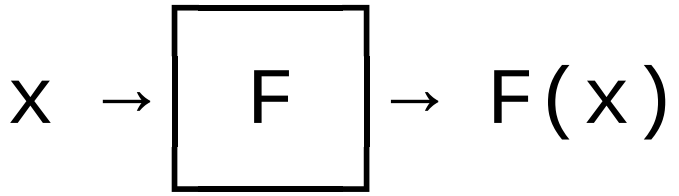
- Se *exprTeste* for verdadeira, o resultado será o valor de *exprV*. Caso contrário, será o valor de *exprF*.

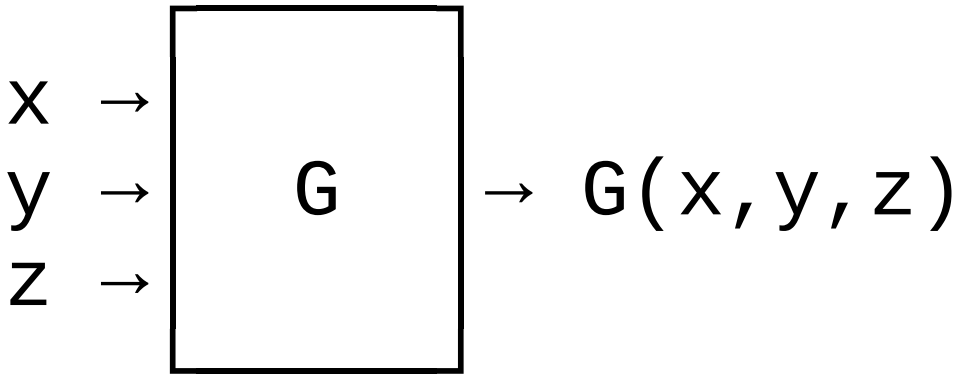
- Exemplos:

```
(numero % 2 == 0) ? "par" : "impar"
diasDeFevereiro = bissexto ? 29 : 28;
```

FUNÇÃO

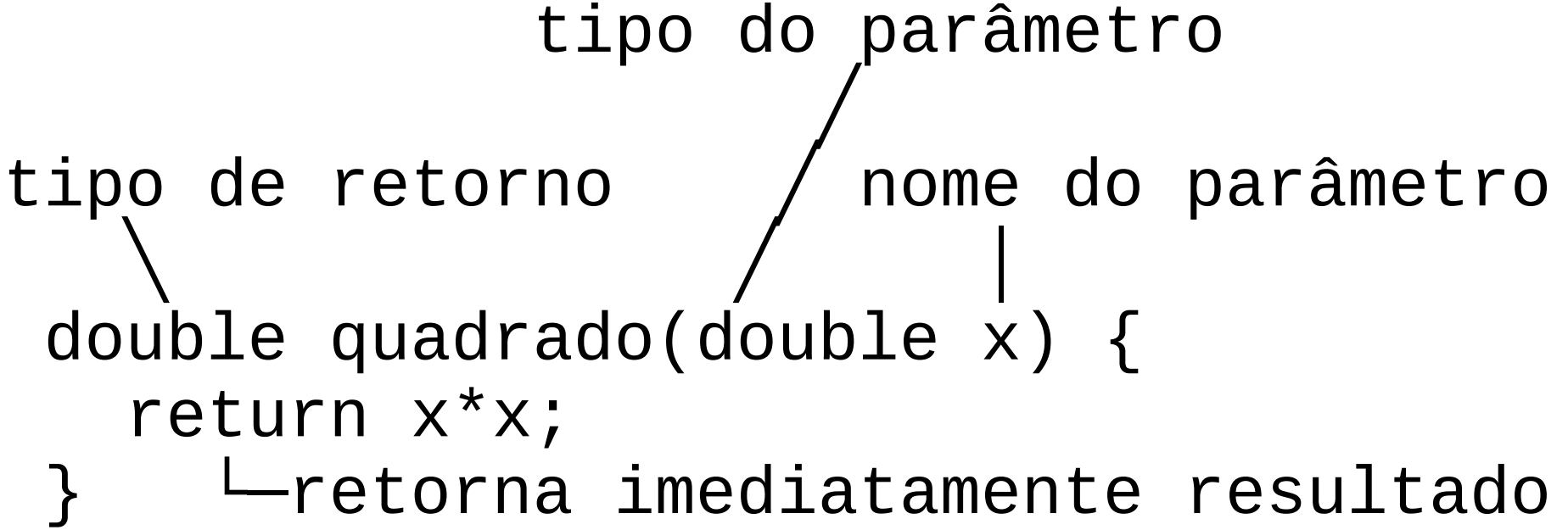
- “Caixa preta” que recebe algo (entrada)
  e produz outra coisa relacionada (saída)





FUNÇÕES em C

- Na linguagem C, os argumentos de em trada de uma função são chamados de parâmetros. Dizemos que a função retorna algo em sua saída.
- Exemplo de definição de uma função:



FUNÇÕES em C - exercícios

- Crie uma função ehBissexto que recebe um ano e retorne verdadeiro se o ano é bissexto ou falso caso contrário.

- Crie uma função discriminante que recebe os coeficientes ***a***, ***b*** e ***c*** de uma equação do segundo grau

$$ax^2 + bx + c = 0$$

e retorne o discriminante dessa equação.

FUNÇÕES em C - exercícios

- Congruência de Zeller

Seja ***K*** os dois últimos algarismos de um ano, ***J*** os demais algarismos, ***q*** o dia do mês e ***m*** o mês no formato: mar=3, abr=4, ..., dez=12, jan=13, fev=14.

A fórmula abaixo dá o dia da semana

$$\left(q + \left\lfloor \frac{13(m+1)}{5} \right\rfloor + K + \left\lfloor \frac{K}{4} \right\rfloor + \left\lfloor \frac{J}{4} \right\rfloor + 5J \right) \bmod 7$$

onde 0 = sáb, 1 = dom, 2 = seg, ...

FUNÇÕES em C - exercícios

Usando a congruência de Zeller, crie uma função diaDaSemana que recebe o dia, mês (de 1 a 12) e ano (com todos os algarismos) e calcula o dia da semana onde 1=dom, 2=seg, ..., 7=sáb.

FUNÇÕES – boas práticas

- ANTES de começar a programar, certifique-se de ter entendido BEM a descrição do problema.

- Suas funções devem se comportar EXATAMENTE como descrito.

- P. ex.: se não está escrito que uma função deve imprimir algo, então ela NÃO DEVE imprimir nada!

FUNÇÕES – boas práticas

- Via de regra, cada função deve **fazer apenas uma coisa, e fazê-la bem**

- P. ex. se não está escrito que uma função vai fazer entrada de valores do usuário, ela **não deve** ler a entrada do usuário.

- Escreva funções **curtas**. Se o código de uma função estiver muito comprido, provavelmente é necessário dividi-la

FUNÇÕES – boas práticas

- Documente suas funções! Minimamente, você deve colocar um comentário antes de cada função, descrevendo:

  * O que ela faz. A primeira frase é crucial.

  * Significado dos parâmetros e valores válidos para eles.

  * Significado do valor de retorno da função.