

Introdução à Computação

Tipos de Dados, Expressões, Operadores e Comandos Básicos

Prof.: Julliano Rosa Nascimento
julliano.rn@gmail.com



Roteiro

- Tipos de Dados: Revisão
- Expressões
- Operadores
- Comandos Básicos
- Estrutura de um Programa



Elementos Fundamentais

Todo trabalho realizado pelo computador é baseado nas informações contidas na **memória**.

Pode ser comparada a um conjunto de caixas **numeradas** e **endereçadas**.



Elementos Fundamentais

Utilizamos endereços físicos e lógicos para recuperar a informação.

- Endereço físico: identificador do circuito de memória.
- Endereço lógico: identificador da variável na memória.

Memória		
Endereço Físico	Endereço Lógico	Informação
500 = 1F4	Nome	Joao
550 = 226	Idade	25
555 = 22B	Altura	1,79
565 = 235	Peso	80



Tipos de Dados

Para que o computador saiba a quantidade de memória a ser reservada para um determinado dado, estes devem ser declarados com seus tipos.

Utilizaremos quatro tipos de dados básicos para representar a informação, na construção de um algoritmo.



Tipos de Dados

inteiro: podem ser positivos, negativos ou nulos.

- Ele tem **15** irmãos.
- O prédio possui **8** andares.

Exemplo em C:

```
int numero_irmaos;  
numero_irmaos = 15;
```



Tipos de Dados

real: podem ser positivos, negativos ou nulos e possuem parte **decimal**.

- Eu tenho **1,76** m de altura.
- Meu plano custa R\$ **88,18**.

Exemplo em C:

```
float altura;  
altura = 1.76;
```



Tipos de Dados

lógico: chamados de booleanos e representados por apenas dois valores; verdadeiro (V) e falso (F).

1 - Verdadeiro
0 - Falso

- A porta pode estar ***aberta*** ou ***fechada***.

Exemplo em C:

```
int porta_aberta;  
porta_aberta = 1;
```



Tipos de Dados

caractere: formado por um único caractere ou uma cadeia deles.

- letras maiúsculas e minúsculas;
- números;
- caracteres especiais (&,#,@,?)
- “Universidade Federal de Goiás”
- “João da Silva”
- “nome@gmail.com”

Exemplo em C:

```
char turma = 'B';  
char nome[30] = "Maria";
```



Tipos de Informações

As informações dos tipos de dados citados podem ser representadas em **C**, como uma constante ou variável.

Constante: um dado é constante é o que não sofre nenhuma modificação no decorrer do tempo.



Constante

- FRASE “Não Fume”
- NOME “Bruna”
- FATOR 1.24
- PI 3.14159

Exemplo em C:

```
#define FATOR 1.24  
#define PI 3.14159
```



Variável

Um dado é classificado como variável quando tem a possibilidade de ser **alterado** em algum instante no decorrer do **tempo**.

Embora possa assumir diferentes valores, só armazena um valor a cada instante.

Área do círculo

$$A = \pi r^2$$



Identificador

Como a variável é representada por um **identificador**, deve ser feita declarando-se da seguinte forma:

```
tipo lista_de_variaveis;
```

Exemplo em C:

```
int idade, num_andares;  
float peso, altura, area;  
char sexo, nome[30], endereço[15];
```



Formação de Identificadores

- Representa **nomes** de variáveis, programas, constantes, rotinas e unidades.
 - Devem começar com um caractere alfabético ou sublinhado (_);
 - Demais alfabéticos, numéricos ou sublinhado;
 - Não são permitidos espaços em branco e caracteres especiais (@, \$, #, %, ...).
 - Não são permitidas palavras reservadas da linguagem.
 - Maiúsculas e minúsculas são caracteres distintos.



Formação de Identificadores

- **Válidos:**

Alpha, x, BJ15, NOTAS, _abc, salario_minimo, media.

- **Inválidos:**

5X, E(13), A:B, Nota/2, AWQ*, P&Q, @123, x-y



Expressões



Expressões Aritméticas

São escritas linearmente usando a notação **matemática**, tendo como resposta sempre um valor **numérico**.

Operador	Função	Exemplos
+	adição	$2+3$, $x+y$
-	subtração	$4-2$, $n-m$
*	multiplicação	$3*4$, $a*b$
/	divisão (inteira ou real, conforme operandos)	$10/2$, $x1/x2$
%	resto da divisão inteira	$16\%3 = 1$



Funções Matemáticas

Além das operações anteriormente citadas, podemos usar nas expressões aritméticas, **funções** da matemática.



Funções Matemáticas

Necessário incluir a biblioteca `<math.h>`

Função	Descrição	Tipo do argumento	Tipo do resultado
<code>sin(x)</code>	seno de x	real ou inteiro	real
<code>cos(x)</code>	cosseno de x	real ou inteiro	real
<code>tan(x)</code>	tangente de x	real ou inteiro	real
<code>pow(x,y)</code>	potência x^y	real ou inteiro	real
<code>sqrt(x)</code>	raiz quadrada de x	real ou inteiro	real
<code>abs(x)</code>	valor absoluto de x	real ou inteiro	real ou inteiro
<code>int(x)</code>	converter x em inteiro	real	inteiro



Funções Matemáticas

O argumento x pode ser uma constante, variável, expressão aritmética ou outra função matemática.

Exemplos:

$$\text{int}(34.886) = 34$$

$$\text{abs}(-27) = 27$$

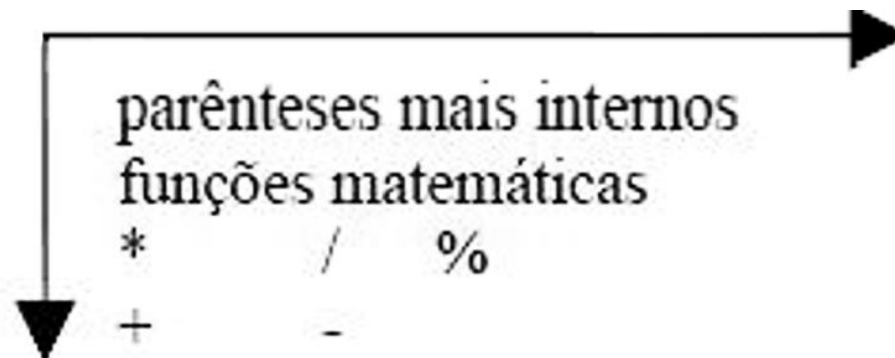
$$\text{pow}(2, 3) = 8$$

$$\text{pow}(\text{sqrt}(9), 3) = 27$$



Funções Matemáticas

Prioridades: guardam uma **hierarquia** entre si.



Funções Matemáticas

Prioridades: guardam uma hierarquia entre si.

$$\begin{aligned}
 \text{a) } & 5 + 9 + 7 + 8/4 \\
 & 5 + 9 + 7 + 2 \\
 & 14 + 7 + 2 \\
 & 21 + 2 \\
 & 23
 \end{aligned}$$

$$\begin{aligned}
 \text{b) } & 1 - 4 * 3/6 - \text{pow}(3,2) \\
 & 1 - 4 * 3/6 - 9 \\
 & 1 - 12/6 - 9 \\
 & 1 - 2 - 9 \\
 & -1 - 9 \\
 & -10
 \end{aligned}$$



Expressões Lógicas

As expressões lógicas trabalham com operadores **relacionais** ou **lógicos**, tendo como resposta um valor lógico (verdadeiro ou falso).

Exemplos:

15 > 3 **V**

0 <= -1 **F**



Operadores



Operadores Relacionais

Operador	Descrição
==	igual a
!=	diferente de
<=	menor ou igual a
>=	maior ou igual a
<	menor que
>	maior que



Operadores Lógicos

Operador	Descrição	Significado
&&	Conjunção	E
	Disjunção	OU
!	Negação	NÃO



Tabela Verdade

Tabela-verdade é o conjunto de **todas** as **possibilidades** combinatórias entre os valores de diversas variáveis lógicas e um conjunto de operadores lógicos.



Tabela Verdade

Seja A a proposição “Hoje é segunda-feira”.

Negação de A :

Hoje **não** é segunda-feira.

A	$\text{NÃO } A$
V	F
F	V



Tabela Verdade

Sejam as proposições:

A: “Hoje é segunda-feira”.

B: “Hoje está chovendo”.

Conjunção de A e B:

Hoje é segunda-feira **e** está chovendo.

A	B	A && B
V	V	V
V	F	F
F	V	F
F	F	F



Tabela Verdade

Sejam as proposições:

A: “Hoje é segunda-feira”.

B: “Hoje está chovendo”.

Disjunção de A e B:

Hoje é segunda-feira **ou** está chovendo.

A	B	A B
V	V	V
V	F	V
F	V	V
F	F	F



Tabela Verdade

Sejam as proposições:

A: “Hoje é segunda-feira”.

B: “Hoje está chovendo”.

Conjunção A e B:

Se é segunda-feira **e** está chovendo, eu fico em casa.

Quando eu fico em casa?



Tabela Verdade

Sejam as proposições:

A: “Hoje é segunda-feira”.

B: “Hoje está chovendo”.

Disjunção A ou B:

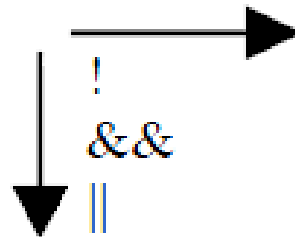
Se é segunda-feira **ou** está chovendo, eu fico em casa.

Quando eu fico em casa?

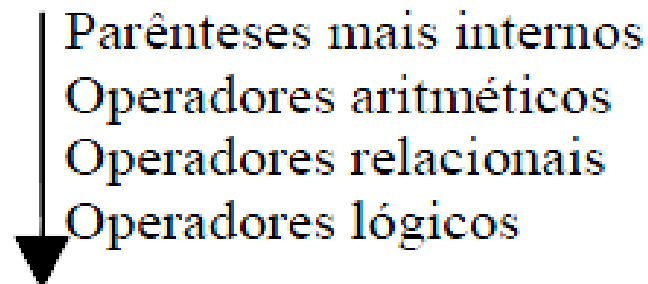


Prioridades

Entre operadores lógicos:



Entre todos os operadores:



Prioridades

`pow(2,4) != 4+2 || 2 + 3 * 5 / 3 % 5 > 0`



Prioridades

`pow(2,4) != 4+2 || 2 + 3 * 5 / 3 % 5 > 0`

`16 != 4+2 || 2 + 3 * 5 / 3 % 5 > 0`

`16 != 4+2 || 2 + 15 / 3 % 5 > 0`

`16 != 4+2 || 2 + 5 % 5 > 0`

`16 != 4+2 || 2 + 0 > 0`

`16 != 4+2 || 2 > 0`

`16 != 6 || 2 > 0`

`V || 2 > 0`

`V || V`

V

potenc.
multiplic.
divisão
resto
soma
soma
diferente de
maior que
disjunção



Comandos Básicos



Atribuição

Utilizado para **atribuir/designar** valor às variáveis.

O tipo de dado deve ser **compatível** com o tipo da variável.



Atribuição

Sintaxe:

```
<identificador> = <expressão>;
```

Onde:

- **<Identificador>** é o nome da variável à qual está sendo atribuído o valor;
- **=** é o símbolo de atribuição;
- **<expressão>** pode ser uma constante, variável ou expressão cujo tipo seja compatível com o da variável de identificador.



Atribuição

Sintaxe:

`<identificador> = <expressão>;`

Exemplos:

```
x = 4;  
turma = 'B';  
cor = "Verde";  
media = (n1+n2)/2;
```

(Lê-se “x recebe 4”)



Comandos de E/S

Tem como finalidade a interrupção do programa para **entrada** ou **saída** de dados.

Em C, a **entrada** de dados é feita com a função *scanf()*;

A **saída** de dados é feita com a função *printf()*;



LEIA

Sintaxe:

```
scanf ("formato", &identificador);
```

Exemplos:

```
scanf ("%d", &inteiro);
```

```
scanf ("%f", &real);
```

```
scanf ("%d/%d/%d", &dia, &mes, &ano);
```



Especificadores de Formato

Especificador	Tipo de Dado
<code>%c</code>	<code>char</code>
<code>%d</code> ou <code>%i</code>	<code>int</code>
<code>%f</code>	<code>float</code>
<code>%s</code>	uma string (sequência de char)



IMPRIMA

Sintaxe:

```
printf ("Mensagem");  
printf ("formato", identificador);
```

Exemplos:

```
printf ("Ola mundo");  
printf ("x = %d", x);  
printf ("x = %d, y = %d", x, y);
```



Estrutura Sequencial

Conjunto de **ações** que serão executadas em sequencia linear de forma **horizontal**, na ordem que forem escritas.

As ações serão seguidas por **ponto e vírgula(;)** .

```
float peso;  
printf("Digite o peso:");  
scanf("%f", &peso);
```



Estrutura de um Programa



Estrutura de um Programa em C

```
1  #include <stdio.h>
2
3  #define PI 3.14 // constantes
4
5  int main(){
6      float raio, area; // declaracao de variaveis
7
8      printf("Digite o valor do raio: ");
9      scanf("%f", &raio);
10     area = PI*raio*raio;
11     printf("A area do circulo e = %f\n", area);
12
13     return 0;
14 }
```



Comentários

Um comentário pode ser utilizado em um programa para que as pessoas possam entendê-lo mais facilmente.

Sintaxe:

```
// Comentário ...
```

Exemplo:

```
int num_pecas; // número de peças  
float salario; // salário do funcionário
```



Exercício 2

Crie um programa que calcule e imprima para o usuário o volume total em metros cúbicos de n colunas de concreto idênticas em formato de cilindro.

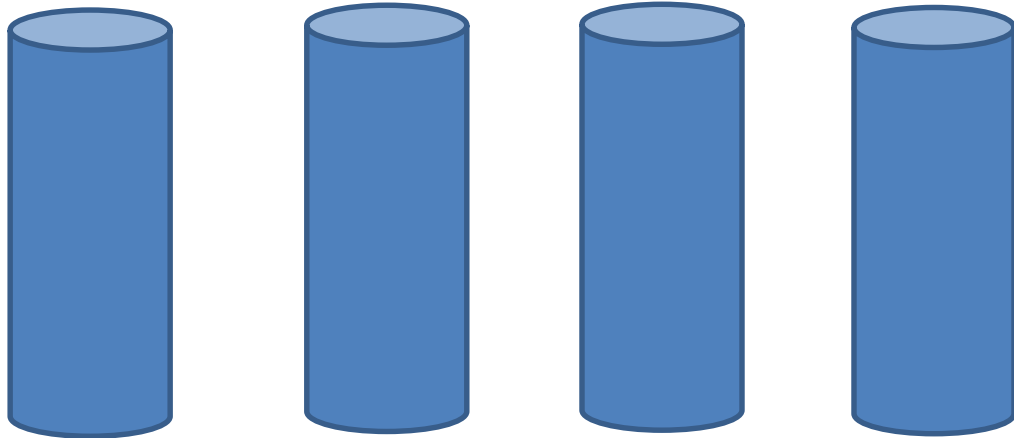
Dado:

$$V_{cil} = \pi r^2 h$$

V_{cil} = volume
do cilindro

r = raio

h = altura



Exercícios 3 e 4

3) Crie um programa que leia o peso (kg) e a altura (m) de uma pessoa e calcule o seu índice de massa corporal (IMC).

$$\text{IMC} = \frac{\textit{peso}}{\textit{altura}^2}$$

4) Crie um programa que leia dois pontos quaisquer do plano cartesiano, $P(x_1, y_1)$ e $Q(x_2, y_2)$, e imprima a distância d entre eles.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Dúvidas?

