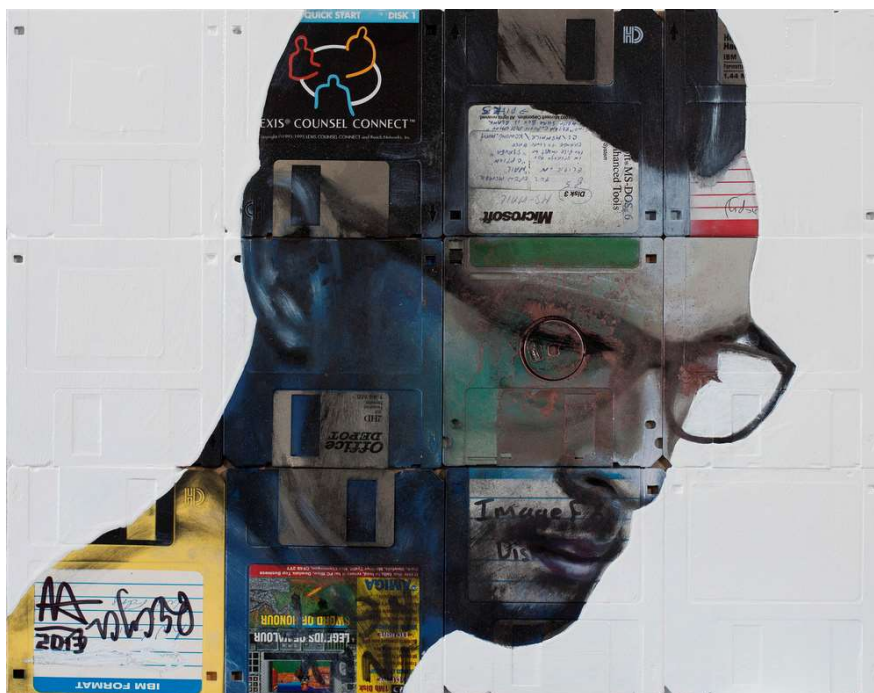


Algoritmos e Programação de Computadores

Disciplina 113476



<http://www.nickgentry.com/>

Prof. Alexandre Zaghetto
<http://alexandre.zaghetto.com>
zaghetto@unb.br

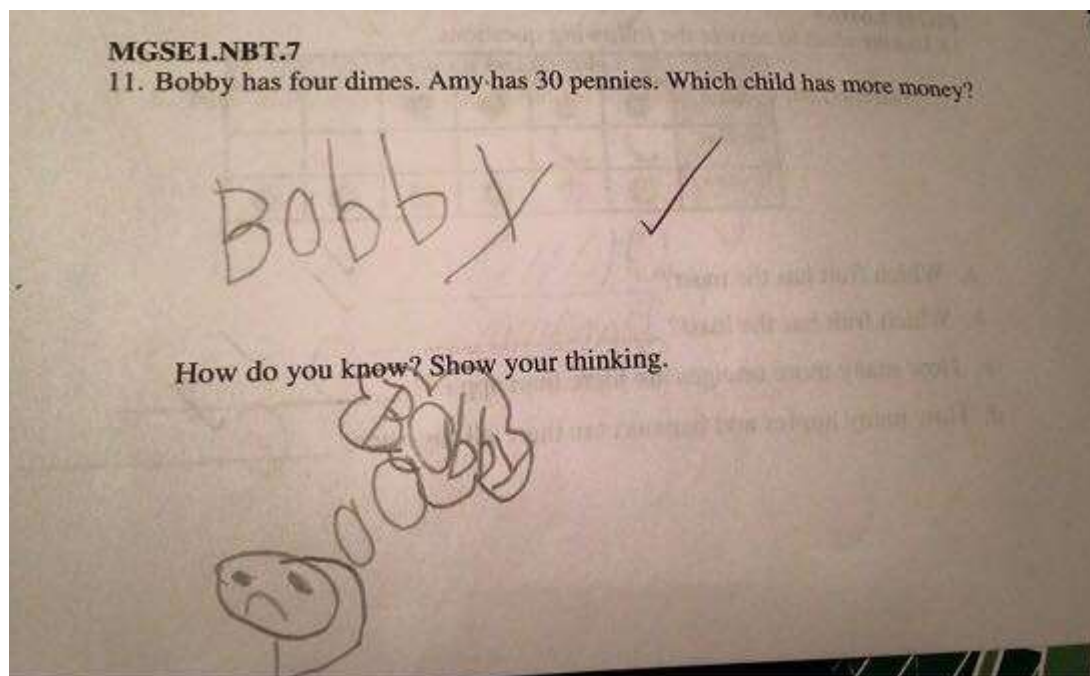
Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

O presente conjunto de *slides* não pode ser reutilizado ou republicado sem a permissão do instrutor.

Módulo 03

O Conceito de Algoritmo

1. O Conceito de Algoritmo



1. O Conceito de Algoritmo

- Conhecimento declarativo:
 - Corpo organizado de informações factuais (o que é).
 - Ex.: **O bolo comum** é a massa de bolo básica para confeitaria ou rechear. É também um excelente acompanhamento para café ou chá.



1. O Conceito de Algoritmo

- Conhecimento imperativo:

- É o que se manifesta da execução de uma tarefa (como fazer).

- Ex.: **Bolo comum:**



1. Bater o açúcar e a manteiga, com a essência de baunilha até branquear.
2. Acrescentar as gemas uma a uma, batendo sempre, até levantar bolhas.
3. Peneirar a farinha, a maizena e o fermento e ir acrescentando pouco a pouco, alternando com o leite, sem parar de bater.
4. Em separado, bater as claras em neve, com a pitada de sal.
5. Misturar as claras delicadamente à mistura
6. Assar em forma untada e polvilhada com farinha de trigo, em forno médio, por aproximadamente 40 minutos.

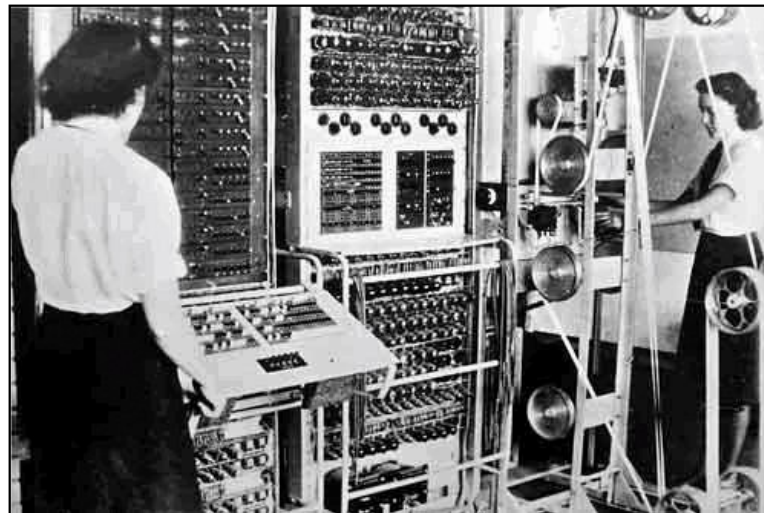
1. O Conceito de Algoritmo



- **Alan Turing** (1912 - 1954), matemático britânico, descreve em 1936 uma máquina teórica, conhecida como **Máquina de Turing**, capaz de realizar qualquer tarefa computável, desde que execute adequadamente uma determinada seqüência de instruções.
- Essa seqüência de instruções é chamada de **algoritmo**.
- Em 1938, foi recrutado pelo departamento de análise criptográfica do governo.
- Conseguiu decifrar o código da máquina de criptografia Enigma, que a Alemanha de Hitler usava para mandar mensagens militares cifradas durante a guerra.

1. O Conceito de Algoritmo

- Graças ao sistema de decodificação que ele criou, o Reino Unido passou a interceptar as mensagens e localizar os submarinos alemães, atacando-os e revertendo o avançar da guerra.
- Mas seu trabalho era secreto, e os feitos de Turing passaram sem aclamação na época.
- Colossus:



1. O Conceito de Algoritmo

- Como homossexual, no início dos anos 1950 foi publicamente humilhado.
- Perdeu o acesso de segurança aos laboratórios onde trabalhava porque, sob a mentalidade da Guerra Fria corrente, homossexuais eram uma brecha na segurança.
- Condenado a terapias à base de estrogênio, o que, de fato, equivalia a castração química e que teve o humilhante efeito secundário de lhe fazer crescer seios.
- Em 8 de junho de 1954, um criado de Turing encontrou-o morto em sua residência em Wilmslow, Cheshire.
- Um exame post-mortem estabeleceu que a causa da morte foi envenenamento por cianeto.

1. O Conceito de Algoritmo

- Um exemplo de Máquina de Turing
 - ✓ A máquina é composta das seguintes partes:
 - a) uma fita dividida em células, potencialmente infinita.
 - b) uma cabeça de leitura/escrita, capaz de:
 - i. Observar apenas uma célula em um dado momento.
 - ii. Ler se a célula tem o número 1 escrito ou se está em branco (0).
 - iii. Escrever ou apagar um símbolo 1.
 - iv. Movimentar-se para a célula imediatamente a esquerda ou direita daquela que está sendo observada.

1. O Conceito de Algoritmo

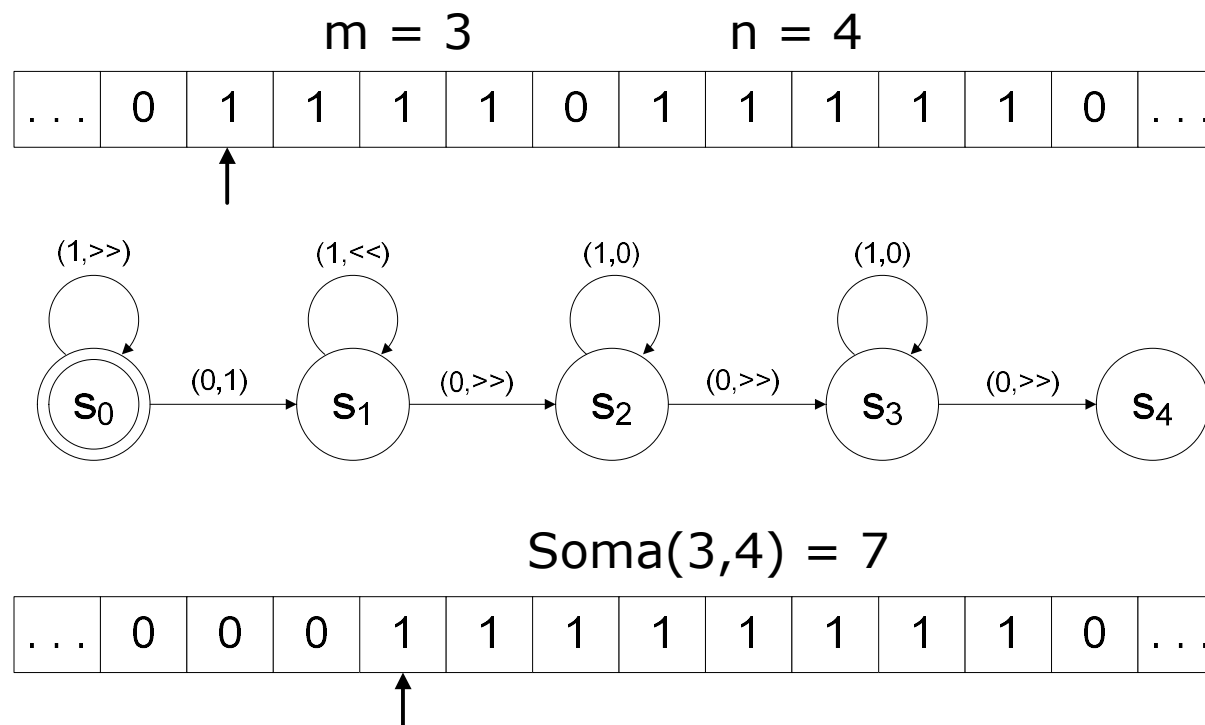
- Um exemplo de Máquina de Turing.
 - ✓ A operação da máquina é determinada pelo estado corrente e pelo símbolo sendo observado, que geram as seguintes operações e um novo estado:
 - a) Escreva o símbolo 1 na célula observada: 1
 - b) Apague qualquer símbolo que apareça na célula observada : 0
 - c) Mova uma célula para a direita da célula observada: >>
 - d) Mova uma célula para a esquerda da célula observada: <<

1. O Conceito de Algoritmo

- Um exemplo de Máquina de Turing.
 - ✓ Assim, uma instrução para a máquina consiste na quádrupla:
$$(s_i, D, Op, s_j)$$
 - ✓ s_i é o estado corrente, $D \in \{0,1\}$ é o símbolo corrente, $Op \in \{0,1,>>,<<\}$ é uma das operações anteriormente descritas e s_j é o novo estado.
 - ✓ Um programa para uma Máquina de Turing é uma coleção finita dessas instruções.
 - ✓ A máquina para quando não há uma transição única (nenhuma ou mais de uma) a ser realizada.

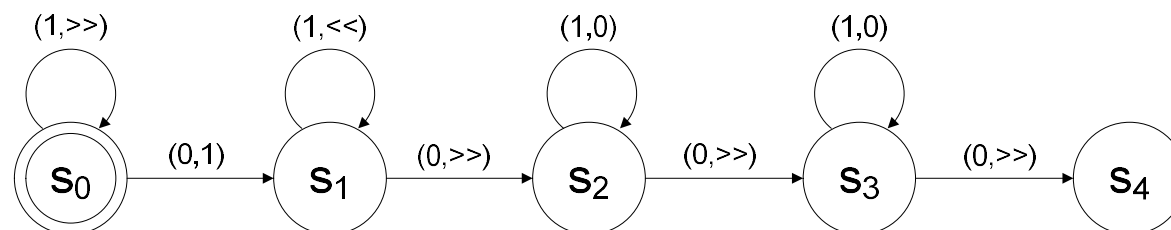
1. O Conceito de Algoritmo

- Um exemplo de Máquina de Turing.
- ✓ Exercício - Máquina de Turing que realiza a soma de dois algarismos: $Soma(m,n) = m+n$:



1. O Conceito de Algoritmo

- Um exemplo de Máquina de Turing.
- ✓ Exercício - Máquina de Turing que realiza a soma de dois algarismos: $Soma(m,n) = m+n$:



$s_0, 1, >>, s_0$
 $s_0, 0, 1, s_1$
 $s_1, 1, <<, s_1$
 $s_1, 0, >>, s_2$
 $s_2, 1, 0, s_2$
 $s_2, 0, >>, s_3$
 $s_3, 1, 0, s_3$
 $s_3, 0, >>, s_4$

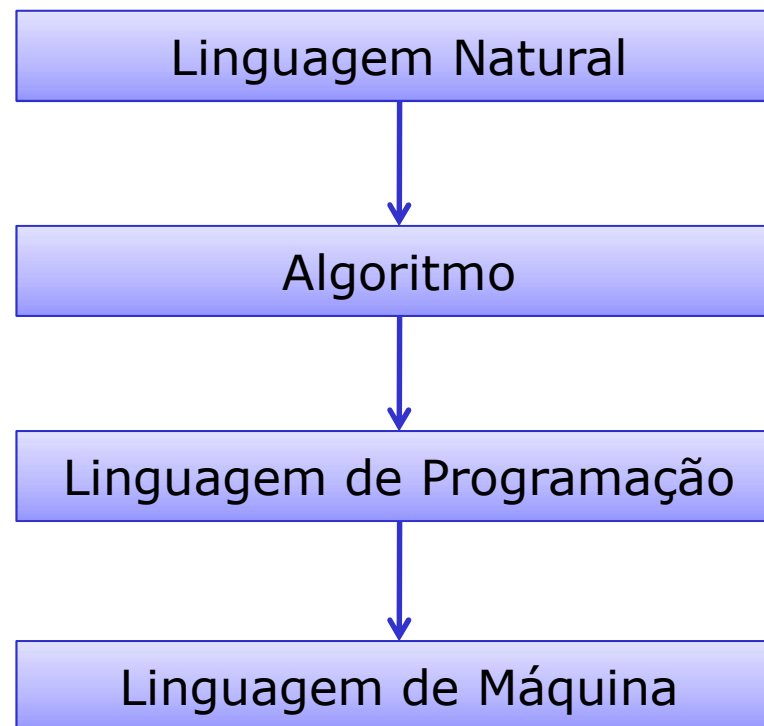
1. O Conceito de Algoritmo

- Um exemplo de Máquina de Turing.
 - ✓ Desafio - Máquina de Turing que realiza a subtração de dois algarismos: $\text{Sub}(m,n) = m - n$?
 - ✓ Para mais informações:
 - <http://plato.stanford.edu/entries/turing-machine/>
 - <http://www.aturingmachine.com/>
 - "On Computable Numbers, with an Application to the *Entscheidungsproblem*", Alan Turing, 1936.

1. O Conceito de Algoritmo

- O conceito central da programação e da ciência da computação é o de **algoritmo**.
- Um algoritmo é a descrição de um padrão de comportamento, expresso em termos de um repertório bem definido e finito de **ações "primitivas"**, das quais damos por certo que elas podem ser executadas.
- Introduziremos de forma intuitiva a noção de algoritmo, motivando ao mesmo tempo as estruturas básicas de controle (**seqüência simples, alternativa e repetição**) como formas de raciocínio "naturais".

1. O Conceito de Algoritmo



2. Algoritmos Não-computacionais

- Seqüência Simples

"traga a cesta com batatas";
"traga a panela do armário";
"descasque as batatas";
"coloque as batatas na panela";
"guarde a cesta";



2. Algoritmos Não-computacionais

- Alternativas

"traga a cesta com batatas";

"traga a panela do armário";

se "roupa é clara" **então**
 "coloque avental";

"descasque as batatas";

"coloque as batatas na panela";

"guarde a cesta";

2. Algoritmos Não-computacionais

- Repetições

"traga a cesta com batatas";
"traga a panela do armário";

se "roupa é clara" **então**
 "coloque avental";

"descasque uma batata";
"coloque a batata na panela";
... (50 vezes)
"descasque uma batata";
"coloque a batata na panela";

"guarde a cesta";



2. Algoritmos Não-computacionais

- Repetições

"traga a cesta com batatas";
"traga a panela do armário";

se "roupa é clara" **então**
 "coloque avental";

se "número de batatas é insuficiente" **então**
 "descasque uma batata";
 "coloque a batata na panela";

se "número de batatas é insuficiente" **então**
 "descasque uma batata";
 "coloque a batata na panela";
 ... (50 vezes)

"guarde a cesta";

2. Algoritmos Não-computacionais

- Repetições

"traga a cesta com batatas";
"traga a panela do armário";

se "roupa é clara" **então**
 "coloque avental";

enquanto "número de batatas é insuficiente" **faça**
 "descasque uma batata";
 "coloque a batata na panela";

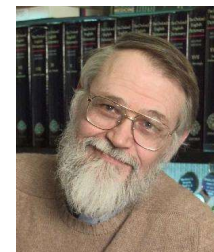
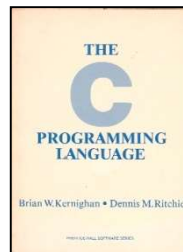
"guarde a cesta";

3. Algoritmos Computacionais

- Um algoritmo computacional é uma seqüência de instruções que manipula dados.
- **Instruções:** comandos que determinam a forma pela qual os dados devem ser tratados.
- **Dados:** informações recolhidas/fornecidas por diversos meios e que serão processadas pelo computador através das instruções.

3. Algoritmos Computacionais

- Até certa parte do curso vamos implementar algoritmos computacionais utilizando pseudocódigo, e as linguagem de programação C e Python.
- Depois de um tempo, vamos abandonar o pseudocódigo e permanecer apenas com o C e o Python.
- Existem várias formas de pseudocódigo.
- A linguagem C foi proposta por **Brian Kernighan** and **Dennis Ritchie** entre o final da década de 1960 e início da década de 1970.





3. Algoritmos Computacionais

- A linguagem foi padronizada pelo ANSI (*American National Standards Institute*) no final da década de 1980 e ficou conhecida como ANSI C.
- Desde então sofreu várias alterações.
- *The Spirit of C:*
 - *Trust the programmer.*
 - *Don't prevent the programmer from doing what needs to be done.*
 - *Keep the language small and simple.*
 - *Provide only one way to do an operation.*
 - *Make it fast, even if it is not guaranteed to be portable.*

3. Algoritmos Computacionais

- As 32 palavras reservadas definidas no ANSI C:

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

3. Algoritmos Computacionais

- **Identificadores:** nomes (rótulos) atribuídos às variáveis, funções e estruturas de dados que são utilizados em algoritmos.
- Regras para formação de identificadores:
 - o primeiro caractere deve ser, obrigatoriamente, uma letra;
 - do segundo caractere em diante são permitidos números e letras;
 - o símbolo de sublinhado (__) pode ser usado para separar nomes compostos;
 - não são permitidos espaços, caracteres acentuados e símbolos especiais na composição do nome de um identificador;
 - palavras reservadas não podem ser usadas (ver slide anterior);
 - há distinção entre maiúsculo e minúsculo.

3. Algoritmos Computacionais

- **Identificadores**

- Exemplos válidos:

DataNascimento
DATA_DE_NASCIMENTO
IDADE
Nota1
TRABALHO2
PESO

- Exemplos inválidos:

3CD
Meu Nome
Idade&
DA*TA
Sílabas

3. Algoritmos Computacionais

- **Variável:** é um espaço reservado na memória do computador para armazenar um determinado tipo de dado.
- Devem receber identificadores para poderem ser referenciadas e modificadas quando necessário.
- Um programa deve conter declarações que especificam de que tipo são as variáveis que ele utilizará e às vezes um valor inicial.
- Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante.

3. Algoritmos Computacionais

- **Variável:**

Memória	
27	Idade
71.5	Massa
ALEXANDRE	Nome
.	
.	
.	

3. Algoritmos Computacionais

- **Tipos básicos** de variável em pseudocódigo

- **inteiro**: número inteiro, negativo, nulo ou positivo.
Ex.: -15, 0, 101

- **real**: número real, negativo, nulo ou positivo. Ex.: -1, -0.5, 0, 5, 9.5

- **caracter**: conjunto de caracteres alfanuméricos. Ex.: "AB", "123", "A123", "CASA"

- **logico**: conjunto de valores FALSO ou VERDADEIRO em proposições lógicas.

3. Algoritmos Computacionais

- **Tipos básicos** de variável em C (por enquanto isso basta, depois veremos muito mais!):

- **int**: número inteiro, negativo, nulo ou positivo. Ex.: -15, 0, 101

- **float**: número real, negativo, nulo ou positivo. Ex.: -1, -0.5, 0, 5, 9.5

- **char**: apenas um único caracter alfanumérico. Ex.: 'A', '3', '2', 'd'

3. Algoritmos Computacionais

- **Estrutura básica** de um algoritmo computacional em pseudocódigo:

```
algoritmo "nome_do_algoritmo"
```

```
// Seção de Declarações
```

```
var
```

```
    <declaração de variáveis>
```

```
// Seção de Comandos
```

```
inicio
```

```
    <comandos>
```

```
finalgoritmo
```

3. Algoritmos Computacionais

- **Estrutura básica** de um algoritmo computacional em C:

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

int main()
{
    // Seção de Declarações
    <declaração de variáveis>

    // Seção de Comandos
    <comandos>

    return 0;
}
```

3. Algoritmos Computacionais

- **Declaração de variáveis** em pseudocódigo:

```
algoritmo "declaravariaveis"  
  
// Seção de Declarações  
var  
inteiro: idade, num_de_filhos  
real: peso, altura  
  
// Seção de Comandos  
inicio  
  
    <comandos>  
  
fimalgoritmo
```

3. Algoritmos Computacionais

- **Declaração de variáveis** em C:

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

int main()
{
    int idade, num_de_filhos;
    float peso, altura;

    // Seção de Comandos
    <comandos>

    return 0;
}
```

4. Algoritmos Computacionais: Operadores

- Operador de atribuição:

Operador	Ação
=	Atribuição

Idade = 30;

a = b = 1.5;

4. Algoritmos Computacionais: Operadores

- Operadores aritméticos:

Operador	Ação
-	Subtração, também menos unário
+	Adição
*	Multiplicação
/	Divisão
%	Módulo da divisão (resto)
--	Decremento
++	Incremento

4. Algoritmos Computacionais: Operadores

- Operadores aritméticos:

Operador	Ação
-	Subtração
+	Adição
*	Multiplicação
/	Divisão

$\text{delta} = b*b - 4*a*c;$



4. Algoritmos Computacionais: Operadores

- Menos unário:

Operador	Ação
-	Menos unário

```
num1= -num;
```


4. Algoritmos Computacionais: Operadores

- Operadores aritméticos:

Operador	Ação
%	Módulo da divisão (resto)

```
num = 17%5;
```

4. Algoritmos Computacionais: Operadores

- Operadores de incremento e decremento:

Operador	Ação
--	Decremento
++	Incremento

a++; ou
a +=1; ou
a = a + 1;
b = ++a; difere de b = a++;

a--; ou
a -=1; ou
a = a - 1;
b = --a; difere de b = a--;

4. Algoritmos Computacionais: Operadores

- Precedência dos operadores aritméticos:

Operador	Precedência
++ --	Mais alta
-	.
* / %	.
+ -	Mais baixa

Operadores do mesmo nível de precedência são avaliados pelo compilador da esquerda para a direita.

ATENÇÃO ➔ Em qualquer caso, os parênteses são sempre prioritários.

4. Algoritmos Computacionais: Operadores

- Operadores aritméticos de atribuição:

Operador	Ação
$a += b$	$a = a + b$
$a -= b$	$a = a - b$
$a *= b$	$a = a * b$
$a /= b$	$a = a / b$
$a \% = b$	$a = a \% b$

$a += 2$; equivale a $a = a + 2$;

5. Algoritmos Computacionais: Funções

- **Algumas funções úteis:**

Função	Ação
<code>pow(a,b)</code>	a^b (a^b)
<code>sqrt(a)</code>	\sqrt{a} ($a^{1/2}$ ou $a^{(1/2)}$)

ATENÇÃO → As funções têm prioridade sobre os outros operadores.

6. Operadores e Funções: Exemplos

- Exemplos:

$$2x^2 - \frac{3x^{(x+1)}}{2} + \frac{\sqrt{x+1}}{x}$$

6. Operadores e Funções: Exemplos

- Exemplos:

$$2x^2 - \frac{3x^{(x+1)}}{2} + \frac{\sqrt{x+1}}{x}$$

$$2*\underline{\text{pow}}(x,2) - 3*\underline{\text{pow}}(x,x+1)/2 + \underline{\text{sqrt}}(x+1)/x$$

6. Operadores e Funções: Exemplos

- Exemplos:

$$2h - \left(\frac{45}{3x} - 4h(3 - h) \right)^{22k}$$

6. Operadores e Funções: Exemplos

- Exemplos:

$$2h - \left(\frac{45}{3x} - 4h(3 - h) \right)^{22k}$$

$2*h - \text{pow}(45/(3*x) - 4*h*(3-h), 22*k)$

6. Operadores e Funções: Exemplos

- Exemplos:

$$a+b+(34+\text{pow}(9,e))/(u-\text{sqrt}(89))$$

6. Operadores e Funções: Exemplos

- Exemplos:

$$a+b+(34+\text{pow}(9,e))/(u-\text{sqrt}(89))$$

$$a + b + \frac{34 + 9^e}{u - \sqrt{89}}$$

6. Operadores e Funções: Exemplos

- Exemplos:

$$(\text{pow}(a+x, 2+w) - 3*a)/2$$

6. Operadores e Funções: Exemplos

- Exemplos:

$$(\text{pow}(a+x, 2+w) - 3*a)/2$$

$$\frac{(a+x)^{2+w} - 3a}{2}$$

7. Entrada e Saída de Dados

- Saída de dados (via monitor):

➤ Em pseudocódigo: **escreva()**

```
escreva("Sua idade é:", idade, "anos.");
```

➤ Em C: **printf()**

```
printf("Sua idade é: %d anos. \n", idade);  
printf("Sua altura é: %f metros. \n", altura);  
printf("A letra é: %c. \n", letra);  
printf("I: %d, A: %f, L: %c. \n", idade, altura, letra);
```

↑
\n → Nova linha

7. Entrada e Saída de Dados

- Entrada de dados (via teclado):

➤ Em pseudocódigo : **leia**()

```
leia(idade);  
leia(altura);  
leia(letra);
```

➤ Em C: **scanf**()

```
scanf("%d", &idade);  
scanf("%f", &altura);  
scanf("%c", &letra);
```

