

# Programação Básica de Computadores

Jordana S. Salamon  
[jssalamon@inf.ufes.br](mailto:jssalamon@inf.ufes.br)

DEPARTAMENTO DE INFORMÁTICA  
CENTRO TECNOLÓGICO  
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO



# Introdução a Linguagem C



nemo

# História

- C é uma linguagem de programação compilada de propósito geral, estruturada, imperativa, procedural, padronizada pela ISO, criada em 1972, por Dennis Ritchie.



Ken Thompson e Dennis Ritchie (da esquerda para direita), os criadores das linguagens B e C, respectivamente.

# Relacionando C com Pseudocódigo



nemo

# Definindo um algoritmo em C

Tradução	
Pseudocódigo	C
Algoritmo MeuPrograma VAR ... Início ... Fim.	main() { ... //comentário /* bloco de comentário */ ... }

- Comentários: entre /\* e \*/ ou depois de //
- Case sensitive*: main ≠ Main ≠ MaIn ≠ mAin ≠ ...
- Uso de “;” ao final de cada instrução

# Função Main

- ▶ É uma função **especial** da linguagem.
- ▶ Sempre o arquivo de execução for executado, esta função será chamada.

```
int main();  
  
int main (void);  
  
void main ();  
  
main ();  
  
int main (int argc, char *argv[])  
  
int main (int argc, char **argv)
```

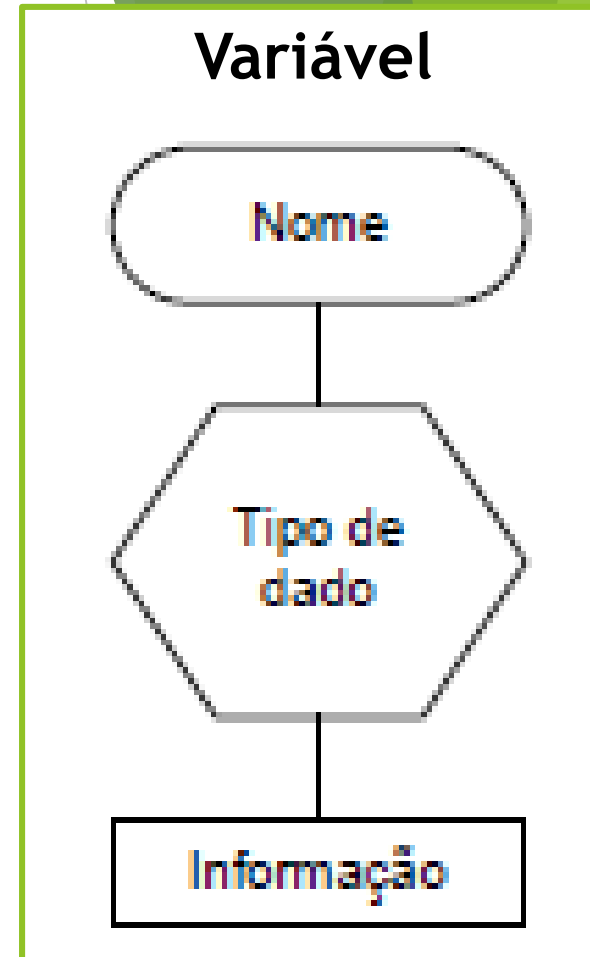


# Tipo de Dados e Declaração de Variáveis

Tradução	
Pseudocódigo	C
Inteiro	int
Real	float
Literal	char
Literal[]	char[]
Lógico	int

Exemplos	
Pseudocódigo	C
<b>VAR</b> a, b, c: <b>inteiro</b> x, y: <b>real</b> pagou: <b>lógico</b> nome: <b>literal</b> [50]	<b>int</b> a, b, c; <b>float</b> x, y; <b>int</b> pagou; <b>char</b> nome[50];

- Uso do “;”
- Tamanho do char
  - Diferença entre char e char[10]
- Não existe lógico em C: usa-se 0 e 1
- short, long, double, long double



# Tabela de tipos de Dados Completa

Tipo	Tamanho em Bytes	Faixa Mínima
char	1	-127 a 127
unsigned char	1	0 a 255
signed char	1	-127 a 127
int	4	-2.147.483.648 a 2.147.483.647
unsigned int	4	0 a 4.294.967.295
signed int	4	-2.147.483.648 a 2.147.483.647
short int	2	-32.768 a 32.767
unsigned short int	2	0 a 65.535
signed short int	2	-32.768 a 32.767
long int	4	-2.147.483.648 a 2.147.483.647
signed long int	4	-2.147.483.648 a 2.147.483.647
unsigned long int	4	0 a 4.294.967.295
float	4	Seis dígitos de precisão
double	8	Dez dígitos de precisão
long double	10	Dez dígitos de precisão



# Instruções Primitivas

As funções scanf e printf fazem parte da biblioteca stdio.h

Tradução	
Pseudocódigo	C
<-	=
Leia	scanf()
Escreva	printf()

Exemplos	
Pseudo	C
a <- 10	a = 10;
Leia x	scanf("%tipo", &x);
Leia nome	scanf("%s", &nome);
Leia nome, idade	scanf("%s%d", &nome, &idade);
Escreva "Olá, mundo!"	printf("Olá, mundo!\n");
Escreva "Nome: ", nome, " e idade: ", idade	printf("Nome: %s e idade: %d\n", nome, idade);

# Códigos de Leitura e Impressão

Tradução	
Código	Tipo
<code>%d</code>	<code>int</code>
<code>%f</code>	<code>float</code>
<code>%c</code>	<code>char</code> //único
<code>%s</code>	<code>char</code> //um ou vários

- `printf()` não quebra linha automaticamente
  - `\n`
- `scanf()` e `printf()` requerem adição de uma biblioteca
  - `#include <stdio.h>`
- `scanf()` necessita do caractere `&` a cada variável

# Exemplo - Pseudocódigo

Faça um algoritmo que leia uma temperatura em Fahrenheit e calcule sua correspondente em Celsius. Ao fim, imprima os dois valores.  $[C = (5 \cdot (F - 32) / 9)]$

Algoritmo Fahrenheit

VAR

tCelsius, tFahrenheit: real;

Início

escreva (“Informe o valor em Fahrenheit: ”)

leia (tFahrenheit)

tCelsius <-  $(5 * (tFahrenheit - 32) / 9)$

escreva (“Valor em Celsius: ”, tCelsius)

Fimalgoritmo

# Exemplo - C

Faça um algoritmo que leia uma temperatura em Fahrenheit e calcule sua correspondente em Celsius. Ao fim, imprima os dois valores.  $[C = (5*(F-32)/9)]$

```
#include <stdio.h>

void main(){
    float tempC, tempF;
    printf("Informe o valor em Fahrenheit: ");
    scanf("%f", &tempF);
    tempC = (5* (tempF - 32) /9);
    printf("Valor em Celsius: %f\n", tempC);
}
```

# Expressões Aritméticas

Tradução						
Pseudo	+	-	*	/	^	√
C	+	-	*	/	pow()	sqrt()

Exemplos	
Pseudo	C
$d \leftarrow \sqrt{(a + (b - c) * 4) / (2^{10})}$	<code>d = sqrt(a + (b - c) * 4) / pow(2, 10);</code>
<code>c &lt;- c + 1</code>	<code>c = c + 1;</code>
<code>c &lt;- c + 1</code>	<code>c++;</code>
<code>c &lt;- c + 3</code>	<code>c += 3;</code>
<code>c &lt;- c - 1</code>	<code>c = c - 1;</code>
<code>c &lt;- c - 1</code>	<code>c--;</code>
<code>c &lt;- c - 3</code>	<code>c -= 3;</code>



# Expressões Lógicas

Tradução					
Pseudo	.V.	.F.	.OU.	.E.	.NÃO.
C	1	0		&&	!

Exemplos	
Pseudo	C
pagou .E. maiorDeldade	pagou && maiorDeldade

# Expressões Relacionais

Tradução						
Pseudo	=	<>	<	<=	>	>=
C	==	!=	<	<=	>	>=

Exemplos	
Pseudo	C
(a = b) .OU. (.NÃO. (b = c) )	(a == b)    (! (b == c) )
(x > y) .E. (z < w)	(x > y) && (z < w)
resposta <> 10	resposta != 10

# Condicionais: Se-Então

Tradução	
Pseudocódigo	C
Se <Condição> Então <Conjunto de Comandos 1> Senão <Conjunto de Comandos 2> Fim_se	if (<Condição>){ <Comandos 1>; }else{ <Comandos 2>; }
Se <Condição> Então <Conjunto de Comandos> Fim_se	if (<Condição>){ <Comandos>; }



# Exemplo de Controle de Fluxo

```
1  #include <stdio.h>
2
3  main() {
4      float N1, N2, Media;
5      printf("Informe a Nota 1: ");
6      scanf("%f", &N1);
7      printf("Informe a Nota 2: ");
8      scanf("%f", &N2);
9      Media = (N1 + N2) / 2;
10     if(Media >= 7){
11         printf("Aprovado\n");
12     }else{
13         printf("Reprovado\n");
14     }
15 }
```



# Exercício

- Construa um algoritmo que receba a idade do usuário e informe se ele tem idade maior ou igual a 18 anos ou não.

```
main() {  
    int x;  
    float y;  
    ...  
}
```

```
if ( Condição ) {  
    <Comandos 1>;  
} else {  
    <Comandos 2>;  
}
```

```
if ( Condição ) {  
    <Comandos>;  
}
```

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

```
int a, b, c;  
float x, y;  
int pagou;  
char  
nome[50];
```

```
printf("Nome: %s e idade: %d\n", nome, idade);
```

# Resolução

```
#include <stdio.h>

main() {
    int idade;
    printf("Digite a idade: ");
    scanf("%d", &idade);
    if(idade >= 18) {
        printf("Maior de idade, com %d anos", idade);
    } else {
        printf("Menor de idade, com %d anos", idade);
    }
}
```

# Controle de Fluxo: Para

Tradução	
Pseudo	Para <var> de <início> até <final> passo <inc> faça <Conjunto de Comandos> Fim_para
C	for (<var> = <início>; <var> <= <final>; <var> = <var> + <inc>){ <Conjunto de Comandos> }

Exemplo	
Pseudo	Para i de 1 até 10 passo de 1 faça <Conjunto de Comandos> Fim_para
C	int i; for (i = 1; i <= 10; i++){ <Conjunto de Comandos> }



# Exercício

- Faça um programa em C que leia duas notas de 4 alunos e diga se o aluno está aprovado ou reprovado. Média = 7.

```
main() {  
    int x;  
    float y;  
    ...  
}
```

```
if ( Condição ) {  
    <Comandos 1>;  
} else {  
    <Comandos 2>;  
}
```

```
int a, b, c;  
float x, y;  
int pagou;  
char  
nome[50];
```

```
scanf("%tipo", &x);
```

```
int i;  
for (i = 1; i <= 10; i++) {  
    <Conjunto de Comandos>  
}
```

```
printf("Nome: %s e idade: %d\n", nome, idade);
```

# Resolução

```
#include <stdio.h>

main() {
    float n1, n2, m;
    int i;

    for(i=1; i<=4; i++) {
        printf("Digite n1 e n2\n");
        scanf("%f %f", &n1, &n2);
        m = (n1+n2)/2;
        if(m<=7) {
            printf("APROVADO\n");
        } else {
            printf("REPROVADO\n");
        }
    }
}
```



# Controle de Fluxo: Enquanto

Tradução	
Pseudo	Enquanto <Condição> faça <Conjunto de Comandos> Fim_enquanto
C	while(<Condição>){ <Conjunto de Comandos> }

Exemplo1		Exemplo2	
Pseudo	i ← 1 Enquanto i ≤ 10 faça <Conjunto de Comandos> i ← i + 1 Fim_enquanto	Leia (x) Enquanto x <> -1 faça <Conjunto de Comandos> Leia (x) Fim_enquanto	
C	int i = 1; while (i ≤ 10) { <Conjunto de Comandos> i++; }	scanf("%d", &x); while (x != -1) { <Conjunto de Comandos> scanf("%d", &x); }	

# Exercício

- Faça um pseudocódigo que some os números digitados pelo usuário, até que seja informado o número -1.

```
main() {  
    int x;  
    float y;  
    ...  
}
```

```
if ( Condição ) {  
    <Comandos 1>;  
} else {  
    <Comandos 2>;  
}
```

```
scanf("%d", &x);  
while (x != -1) {  
    <Conjunto de Comandos>  
    scanf("%d", &x);  
}
```

```
scanf("%tipo", &x);
```

```
printf("Nome: %s e idade: %d\n", nome, idade);
```

```
int a, b, c;  
float x, y;  
int pagou;  
char  
nome[50];
```





# Biblioteca Matemática e Strings

# Math - Funções úteis

Função	Semântica
#include <math.h>	
<code>sqrt(valorReal)</code>	raiz quadrada
<code>pow(valorReal1, valorReal2)</code>	potência
<code>ceil(valorReal)</code>	teto (arredondar para cima)
<code>floor(valorReal)</code>	piso (arredondar para baixo)

## Raiz - SQRT

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

main() {
    int x, raiz;
    printf("Digite o valor: ");
    scanf("%d", &x);
    raiz = sqrt(x);
    printf("O valor da raiz quadrada de %d eh: %d\n\n", x, raiz);
}
```

## Potencia - POW

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

main() {
    int x, y, pot;
    printf("Digite o valor da base: ");
    scanf("%d", &x);
    printf("Digite o valor da potencia: ");
    scanf("%d", &y);
    pot = pow(x,y);
    printf("O valor da potencia de %d por %d eh: %d\n\n", x, y, pot);
}
```



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

main() {
    float x, ab, ac;
    printf("Digite o valor: ");
    scanf("%f", &x);
    ab = floor(x);
    ac = ceil(x);
    printf("O valor arredondado para cima de %.2f eh: %.2f\n", x, ac);
    printf("O valor arredondado para baixo de %.2f eh: %.2f\n\n", x, ab);
}
```



# Caractere - O que é?

- ▶ Representa um símbolo que pode representar o conjunto de numéricos (0...9), alfabéticos (a...z, A...Z) e especiais (! @ # \$ % & \*).
- ▶ Em C, um caractere é representado pelo tipo **char** e é delimitado em seu início e término como o caractere aspas simples ('');

```
char c, n, e;  
c = 'a';  
n = '0';  
e = '#';
```



# String - O que é?

- ▶ Representa uma cadeia (sequencia) de caracteres.
- ▶ Em C, uma string é representado pelo tipo `char[]` e é delimitado em seu inicio e termino como o caractere aspas simples (");
- ▶ Seu tamanho é dados pelo número de caracteres.

Exemplo:

- ▶ "palavra"; (Tamanho 7)
- ▶ "123"; (Tamanho 3)
- ▶ "&hg\$"; (Tamanho 4)

```
char c[7] = "palavra";  
char n[3] = "abc";  
char e[4] = "&hg$";
```

- ▶ Obs:
  - ▶ 'a' é diferente de "a"
  - ▶ Uma string não pode receber um caractere simples (nome = 'X'; causa erro).

# String - Leitura

- Para a leitura de uma **string** pelo teclado, utilizamos o código %s. Contudo diferente de todos os outros tipos, em strings não utilizado o **&** no scanf.

```
int n;  
scanf("%d", &n);  
float y;  
scanf("%f", &y);  
char c;  
scanf("%c", &c);  
char nome[10];  
scanf("%s", nome); // sem o &
```

# String - Tamanho

- ▶ Caso seja declarado uma string de tamanho 10 (char s[10];) e for digitado um texto com mais de 10 caracteres, será armazenado apenas os 10 primeiros.
- ▶ Caso seja declarado uma string de tamanho 7, o programa funcionará normalmente, por que ao final de cada cadeia de caracteres existe o símbolo '\0', que indica o final da string.

```
char s[10] = "CASA";
```

S =

'C'	'A'	'S'	'A'	'\0'
-----	-----	-----	-----	------





# String - Limitações

- Uma string não pode receber outra diretamente. Tentar fazer isso gera um erro.

```
char s1[10], s2[10];  
s1 = "CASA1"; //erro  
s2 = s1; //erro
```

- Aplicar a operação de soma entre duas strings não é uma operação válida para concatenação.

```
char s1[10] = "CA", s2[10] = "SA";  
s1 = s1 + s2; //erro
```



# String - Funções úteis

#include <string.h>	
Função	Semântica
<code>strcpy(strDestino, strOrigem)</code>	atribuir uma <i>string</i> a outra
<code>strcat(strDestino, strOrigem)</code>	concatenar uma <i>string</i> a outra
<code>strlen(str)</code>	saber o tamanho de uma <i>string</i>
<code>strcmp(str1, str2)</code>	comparar uma <i>string</i> com outra <ul style="list-style-type: none"><li>• retorna 0 se iguais,</li><li>• &lt; 0 se <code>str1 &lt; str2</code></li><li>• &gt; 0 se <code>str1 &gt; str2</code></li></ul>

# String - Funções úteis

Exemplos	
Pseudocódigo	C
<code>nome &lt;- "Joaquim"</code>	<code>strcpy(nome, "Joaquim")</code>
<code>nomeCompleto &lt;- nome</code> <code>nomeCompleto &lt;- nomeCompleto +</code> <code>sobrenome</code>	<code>strcpy(nomeCompleto, nome)</code> <code>strcat(nomeCompleto,</code> <code>sobrenome)</code>
<code>nome1 &lt; nome2</code>	<code>strcmp(nome1, nome2) &lt; 0</code>
<code>nome1 == nome2</code>	<code>strcmp(nome1, nome2) == 0</code>
<code>nome1 &gt; nome2</code>	<code>strcmp(nome1, nome2) &gt; 0</code>



# Exercícios Loop e Condicional em C

1. Faça um programa em C que leia dois pontos (x1, y1 e x2, y2) e calcule a distância entre os pontos. Utilize as funções `sqrt(<numero>)` e `pow(<base>, <potencia>)`;

$$D = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$$

2. Faça um programa em C que leia um nome, e verifique quantos caracteres o nome possui. Considerando que nome pequenos possuem caracteres  $\leq 5$ , médios  $5 < \text{caracteres} \leq 8$  e grandes caracteres  $> 8$ , informe o tamanho do nome ("pequeno", "médio" ou "grande"). Utilize a função `strlen(<string>)`;
3. Faça um programa em C que exiba a tabuada de um número ***n***, de ***n***\*1 a ***n***\*10.
4. Faça um pseudocódigo que calcule a média de uma quantidade indeterminada de números pares positivos, até que seja informado o número 0.

# Resolução - Ex1

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

main() {
    float x1, x2, y1, y2, d;
    printf("Digite o primeiro ponto: ");
    scanf("%f %f", &x1, &y1);
    printf("Digite o segundo ponto: ");
    scanf("%f %f", &x2, &y2);

    d = sqrt(pow((x2-x1),2) + (pow((y2-y1),2)));
    printf("\nA distancia eh: %.2f\n",d);
}
```



# Resolução - Ex2

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

main() {
    char nome[3];
    int tam;

    printf("Digite seu nome: \n");
    scanf("%s", nome);

    tam = strlen(nome);

    if(tam <= 5) {
        printf("Pequeno");
    }
    if(tam > 5 && tam <=8) {
        printf("Medio");
    }
    if(tam >8) {
        printf("Grande");
    }
}
```



# That's all Folks!



nemo