

Comandos Básicos em C

Algoritmos e Estruturas de Dados I

Prof. Lucas Astore

Prof. Cristiano Rodrigues

Variáveis na linguagem C

Dados

Dado – toda e qualquer tipo de informação manipulada por um programa

Tipo de Dado – representação computacional da informação. Define como armazená-la e como interpretá-la

Tipos Básicos – mais utilizados. Podem ser numéricos, lógicos, literais ou caracteres

Exemplos:

2, 10, 381

-4, -215

10.21, -25.63

true, false

“Esse é um exemplo”

'c', '*', ' ', '1'

ASCII TABLE

Como um computador –
que é uma máquina
essencialmente
matemática
(entende 0s e 1s)
– entende e
exibe esses
dados?

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Sinais gráficos (imprimíveis)

Bin	Oct	Dec	Hex	Sinal
0010 0000	040	32	20	(espaço)
0010 0001	041	33	21	!
0010 0010	042	34	22	"
0010 0011	043	35	23	#
0010 0100	044	36	24	\$
0010 0101	045	37	25	%
0010 0110	046	38	26	&
0010 0111	047	39	27	'
0010 1000	050	40	28	(
0010 1001	051	41	29)
0010 1010	052	42	2A	*
0010 1011	053	43	2B	+
0010 1100	054	44	2C	,
0010 1101	055	45	2D	-
0010 1110	056	46	2E	.
0010 1111	057	47	2F	/
0011 0000	060	48	30	0
0011 0001	061	49	31	1
0011 0010	062	50	32	2
0011 0011	063	51	33	3
0011 0100	064	52	34	4
0011 0101	065	53	35	5
0011 0110	066	54	36	6
0011 0111	067	55	37	7
0011 1000	070	56	38	8
0011 1001	071	57	39	9
0011 1010	072	58	3A	:
0011 1011	073	59	3B	;
0011 1100	074	60	3C	<
0011 1101	075	61	3D	=
0011 1110	076	62	3E	>
0011 1111	077	63	3F	?

Bin	Oct	Dec	Hex	Sinal
0100 0000	100	64	40	@
0100 0001	101	65	41	A
0100 0010	102	66	42	B
0100 0011	103	67	43	C
0100 0100	104	68	44	D
0100 0101	105	69	45	E
0100 0110	106	70	46	F
0100 0111	107	71	47	G
0100 1000	110	72	48	H
0100 1001	111	73	49	I
0100 1010	112	74	4A	J
0100 1011	113	75	4B	K
0100 1100	114	76	4C	L
0100 1101	115	77	4D	M
0100 1110	116	78	4E	N
0100 1111	117	79	4F	O
0101 0000	120	80	50	P
0101 0001	121	81	51	Q
0101 0010	122	82	52	R
0101 0011	123	83	53	S
0101 0100	124	84	54	T
0101 0101	125	85	55	U
0101 0110	126	86	56	V
0101 0111	127	87	57	W
0101 1000	130	88	58	X
0101 1001	131	89	59	Y
0101 1010	132	90	5A	Z
0101 1011	133	91	5B	[
0101 1100	134	92	5C	\
0101 1101	135	93	5D]
0101 1110	136	94	5E	^
0101 1111	137	95	5F	_

Bin	Oct	Dec	Hex	Sinal
0110 0000	140	96	60	`
0110 0001	141	97	61	a
0110 0010	142	98	62	b
0110 0011	143	99	63	c
0110 0100	144	100	64	d
0110 0101	145	101	65	e
0110 0110	146	102	66	f
0110 0111	147	103	67	g
0110 1000	150	104	68	h
0110 1001	151	105	69	i
0110 1010	152	106	6A	j
0110 1011	153	107	6B	k
0110 1100	154	108	6C	l
0110 1101	155	109	6D	m
0110 1110	156	110	6E	n
0110 1111	157	111	6F	o
0111 0000	160	112	70	p
0111 0001	161	113	71	q
0111 0010	162	114	72	r
0111 0011	163	115	73	s
0111 0100	164	116	74	t
0111 0101	165	117	75	u
0111 0110	166	118	76	v
0111 0111	167	119	77	w
0111 1000	170	120	78	x
0111 1001	171	121	79	y
0111 1010	172	122	7A	z
0111 1011	173	123	7B	{
0111 1100	174	124	7C	
0111 1101	175	125	7D	}
0111 1110	176	126	7E	~

Exemplo: O que significa o código abaixo?

110 117 118 101 109

Sinais gráficos (imprimíveis)

Bin	Oct	Dec	Hex	Sinal
0010 0000	040	32	20	(espaço)
0010 0001	041	33	21	!
0010 0010	042	34	22	"
0010 0011	043	35	23	#
0010 0100	044	36	24	\$
0010 0101	045	37	25	%
0010 0110	046	38	26	&
0010 0111	047	39	27	'
0010 1000	050	40	28	(
0010 1001	051	41	29)
0010 1010	052	42	2A	*
0010 1011	053	43	2B	+
0010 1100	054	44	2C	,
0010 1101	055	45	2D	-
0010 1110	056	46	2E	.
0010 1111	057	47	2F	/
0011 0000	060	48	30	0
0011 0001	061	49	31	1
0011 0010	062	50	32	2
0011 0011	063	51	33	3
0011 0100	064	52	34	4
0011 0101	065	53	35	5
0011 0110	066	54	36	6
0011 0111	067	55	37	7
0011 1000	070	56	38	8
0011 1001	071	57	39	9
0011 1010	072	58	3A	:
0011 1011	073	59	3B	;
0011 1100	074	60	3C	<
0011 1101	075	61	3D	=
0011 1110	076	62	3E	>
0011 1111	077	63	3F	?

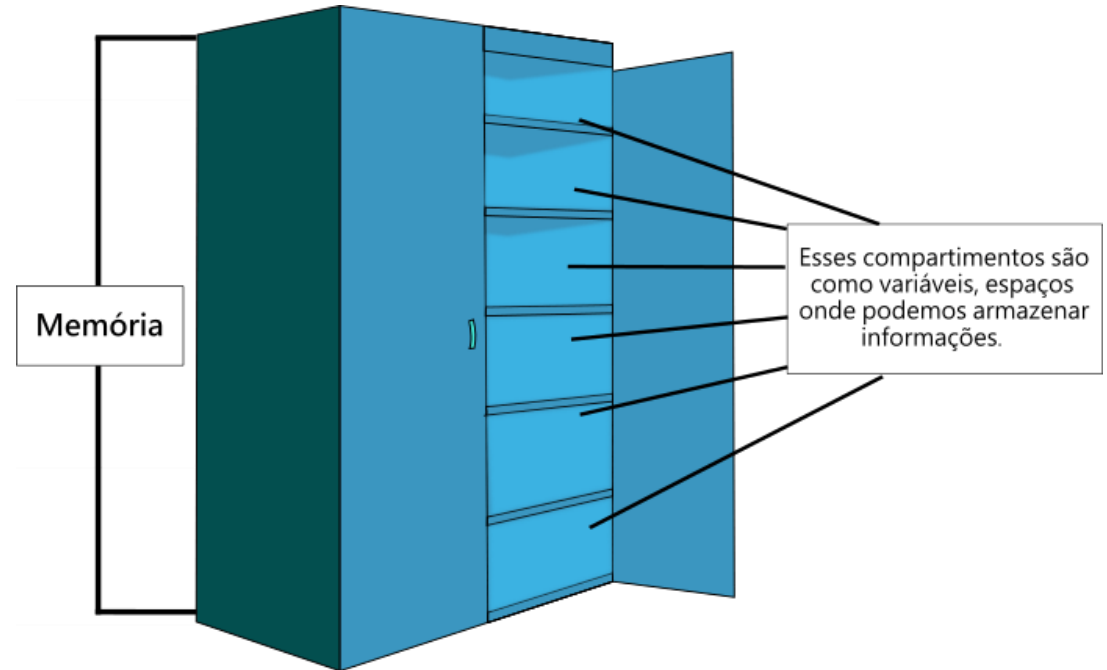
Bin	Oct	Dec	Hex	Sinal
0100 0000	100	64	40	@
0100 0001	101	65	41	A
0100 0010	102	66	42	B
0100 0011	103	67	43	C
0100 0100	104	68	44	D
0100 0101	105	69	45	E
0100 0110	106	70	46	F
0100 0111	107	71	47	G
0100 1000	110	72	48	H
0100 1001	111	73	49	I
0100 1010	112	74	4A	J
0100 1011	113	75	4B	K
0100 1100	114	76	4C	L
0100 1101	115	77	4D	M
0100 1110	116	78	4E	N
0100 1111	117	79	4F	O
0101 0000	120	80	50	P
0101 0001	121	81	51	Q
0101 0010	122	82	52	R
0101 0011	123	83	53	S
0101 0100	124	84	54	T
0101 0101	125	85	55	U
0101 0110	126	86	56	V
0101 0111	127	87	57	W
0101 1000	130	88	58	X
0101 1001	131	89	59	Y
0101 1010	132	90	5A	Z
0101 1011	133	91	5B	[
0101 1100	134	92	5C	\
0101 1101	135	93	5D]
0101 1110	136	94	5E	^
0101 1111	137	95	5F	_

Bin	Oct	Dec	Hex	Sinal
0110 0000	140	96	60	`
0110 0001	141	97	61	a
0110 0010	142	98	62	b
0110 0011	143	99	63	c
0110 0100	144	100	64	d
0110 0101	145	101	65	e
0110 0110	146	102	66	f
0110 0111	147	103	67	g
0110 1000	150	104	68	h
0110 1001	151	105	69	i
0110 1010	152	106	6A	j
0110 1011	153	107	6B	k
0110 1100	154	108	6C	l
0110 1101	155	109	6D	m
0110 1110	156	110	6E	n
0110 1111	157	111	6F	o
0111 0000	160	112	70	p
0111 0001	161	113	71	q
0111 0010	162	114	72	r
0111 0011	163	115	73	s
0111 0100	164	116	74	t
0111 0101	165	117	75	u
0111 0110	166	118	76	v
0111 0111	167	119	77	w
0111 1000	170	120	78	x
0111 1001	171	121	79	y
0111 1010	172	122	7A	z
0111 1011	173	123	7B	{
0111 1100	174	124	7C	
0111 1101	175	125	7D	}
0111 1110	176	126	7E	~

Variáveis

Variável – local de armazenamento reservado na memória RAM (volátil) de um computador em tempo de execução, associado a um **tipo** e a um **nome**. Com o término do programa, a variável é desalocada da memória.

O nome será escolhido pelo desenvolvedor e deve atender a algumas regras. Sugere-se que comece com minúscula ou '_' e todas as palavras que compõe o nome com maiúsculas (CamelCase).

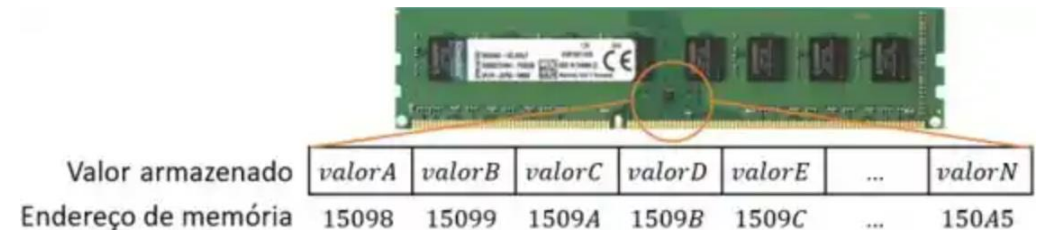


Variáveis

Variável – local de armazenamento reservado na memória RAM (volátil) de um computador em tempo de execução, associado a um **tipo** e a um **nome**. Com o término do programa, a variável é desalocada da memória.

O nome será **escolhido** pelo desenvolvedor e deve atender a algumas regras.

Sugere-se que comece com minúscula ou '_' e todas as palavras que compõe o nome com maiúsculas (CamelCase).



Variáveis

1. Diferente de palavras reservadas
2. Não pode ter símbolos, exceto o *underscore* '_'
3. Deve começar por uma letra ou pelo *underscore* '_'
4. Não podem existir nomes repetidos em um mesmo escopo
5. Minúsculas e maiúsculas são consideradas diferentes (*case sensitive*)
6. Não podem ser acentuados ou utilizar o cedilha 'ç'

Exemplos:

- NomeCompleto, nomeCompleto
- dataNascimento, data_nasc
- saldoConta
- ID

Variáveis - “Clean code”

1. Use nomes que revelem seu propósito
`int d; // tempo decorrido em dias`
2. Evite informações erradas
Uma variável chamada `accountList` que não é uma list
3. Faça distinções significativas
`productInfo` e `productData`
4. Use nomes pronunciáveis
`genymdhms` – (generation date, year, month, day, hour, minute, second)
5. Use nomes passíveis de busca
cuidado com variáveis de uma só letra
6. Evite codificações
garante que mais pessoas possam entender o código

Variáveis

Declaração de Variável – é o comando que efetivamente “cria” a variável, reservando-a em memória e associando nome e tipo a ela. O local onde a variável foi armazenada é identificado internamente no computador pelo *endereço de memória*.

Atribuição – armazena dado na variável substituindo conteúdo anterior pelo indicado

Tipos Básicos em C/C++

- `int` (inteiro, sem casa decimal),
- `float/double` (ponto flutuante, com casa decimal),
- `bool` (lógico),
- `char` (caracter)

Formato

`<tipo> <nomeEscolhido> = dadoInicial;`
(opcional)

Exemplos

```
int quantidade, portas = 4, idade;  
double velocidade = 5.0;  
float aceleracao = 2.0;  
bool carregado = true;
```

Tipo	Bits	Bytes	Escala
char	8	1	128 a 127
int	32	4	-2.147.483.648 a 2.147.483.647 (ambientes de 32 bits)
short	16	2	-32.765 a 32.767
long	32	4	-2.147.483.648 a 2.147.483.647
unsigned char	8	1	0 a 255
unsigned	32	4	0 a 4.294.967.295 (ambientes de 32 bits)
unsigned long	32	4	0 a 4.294.967.295
unsigned short	16	2	0 a 65.535
float	32	4	$3,4 \times 10^{-38}$ a $3,4 \times 10^{38}$
double	64	8	$1,7 \times 10^{-308}$ a $1,7 \times 10^{308}$
long double	80	10	$3,4 \times 10^{-4932}$ a $3,4 \times 10^{4932}$
void	0	0	nenhum valor

Variáveis

Conversão de tipos – é a mudança da forma de representação de um tipo para outro

1. **Conversão implícita** – ocorre automaticamente para alguns tipos. Normalmente, refere-se a um subconjunto atribuído a uma variável com tipo que o englobe
2. **Conversão explícita** – necessária quando existe o risco de perder informação (precisão) na conversão.

Exemplos em C/C++

```
float aceleracao = 2;
```

```
double velocidade = 5.0;
```

```
char letra = 'a';
```

```
int quantidade = (int) velocidade; //  
casting
```

Variáveis

Constante – informação armazenada em um primeiro momento não pode ser alterada posteriormente

O nome segue as regras dos identificadores das variáveis. Sugere-se o uso de maiúsculas apenas.

Define constantes em C

`define, const`

Exemplos

```
#define VELOCIDADE 5.0  
const double VELOCIDADE = 5.0;
```

Entrada e saída no console

Saída

A saída padrão de um programa desenvolvido para Console é a tela.

Como os dados armazenados na memória do computador são apenas 0's e 1's, o tipo indica como representar a informação. Mas a disposição dela na saída também pode ser formatada.

Em C – printf (stdio.h)

- `printf(...);`

Escreve a informação entre parênteses a partir da posição atual do cursor no Console

- `printf("...%?...%?...", var1, var2, ...);`

Nas interrogações os conteúdos das variáveis na ordem indicada é integrado ao texto (conjunto de caracteres).

Saída formatada

Especificador	Tipo
%d	int
%.xf	float (opcional x casas decimais)
%.xlf	double (opcional x casas decimais)
%s	char[]
%b	bool
%c	char

Indica-se com o especificador a posição na string de saída na qual o dado será formatado, sendo que a ordem dos especificadores corresponde à ordem das variáveis indicadas após a string

```
printf ("O carro possui %d portas e sua velocidade é %.2lf km/h", quantidade, veloc);
```

Códigos de formatação para printf()	Significado
%c	Caractere simples.
%d	Inteiro decimal com sinal.
%i	Inteiro decimal com sinal.
%e	Notação científica (e minúsculo).
%E	Notação científica (E maiúsculo).
%f	Ponto flutuante em decimal.
%g	Usa %e ou %f, o que for menor.
%G	Usa %E ou %f, o que for menor.
%o	Inteiro octal sem sinal.
%s	String de caracteres.
%u	Inteiro decimal sem sinal.
%x	Inteiro hexadecimal sem sinal (letras minúsculas).
%X	Inteiro hexadecimal sem sinal (letras maiúsculas).
%p	Ponteiro (endereço).
%n	Ponteiro inteiro.
%%	Imprime um caractere %.

Caracteres especiais

Dentro da string (entre os aspas), pode-se incluir após uma ‘\’ caracteres que assumem funções especiais

\n Nova linha

\t Tabulação

\\ Exibe a \

\” Exibe o “

Exemplos

```
printf("Digite um número: ");
```

```
printf("Digite um número: \n");
```

```
printf("Os números %d e %.2f foram digitados", semCasa, com2Casas);
```

Concatenação String

Uma string deve começar e terminar em uma mesma linha, ou seja, o abre aspas e o fecha aspas devem estar na mesma linha.

Para textos maiores, pode-se utilizar mais de uma string em sequência, delimitadas por aspas.

Em C

```
printf("Esse é um texto "  
      "muito extenso\n");
```

Esse é um texto muito extenso

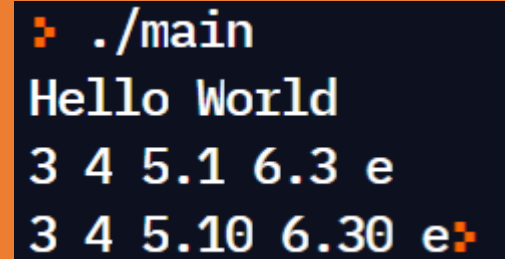
—

Entrada – scanf (stdio.h)

A entrada padrão de um programa desenvolvido para Console é o teclado.

O comando scanf possui o seguinte formato geral:

```
scanf ("%? %? %?", &var1, &var2, &var3);
```

A terminal window with an orange border and dark background. It shows the execution of a program. The prompt is './main'. The first line of input is 'Hello World'. The second line of input is '3 4 5.1 6.3 e'. The third line of input is '3 4 5.10 6.30 e'.

```
./main  
Hello World  
3 4 5.1 6.3 e  
3 4 5.10 6.30 e
```

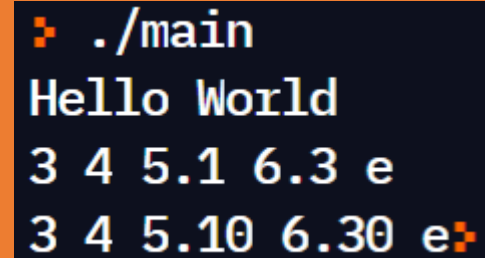
Entrada – scanf (stdio.h)

Em C:

```
#include <stdio.h>

int main(void) {
    int a, b;
    double val;
    float val2;
    char c;

    printf("Hello"
    " World\n");
    scanf("%d %d %lf %f %c",&a, &b,&val,&val2, &c);
    printf("%d %d %.2lf %.2f %c",a, b,val,val2,c);
    return 0;
}
```

A terminal window with an orange border and dark background. It shows the execution of a C program. The first line is a prompt followed by './main'. The second line is the output 'Hello World'. The third line is a prompt followed by '3 4 5.1 6.3 e'. The fourth line is the output '3 4 5.10 6.30 e' followed by a prompt.

```
➤ ./main
Hello World
3 4 5.1 6.3 e
3 4 5.10 6.30 e➤
```

Operadores e expressões

Operadores Matemáticos

Símbolo	Exemplo	Operação
+	2 + 3 X + 3	Soma
-	3 - 5	Subtração
*	2 * 6	Multiplicação
/	3.0 / 2	Divisão
%	3 % 2	Resto da divisão

Operadores Atribuição

Símbolo	Exemplo	Equivale
+= -= *= /=	x += y x *= 9	x = x + y x = x * 9
%= ++ --		
	x++ y--	x = x + 1 y = y - 1

Operadores Relacionais (Binários)

Símbolo	Exemplo	Operação
==	x == y x == true nome == "Pedro"	Igualdade
!=	presente != true	Diferença
<	idade < 17	Menor
<=	valorCompra <= 100	Menor igual
>	altura > 1.70	Maior
>=	idade >= 18	Maior igual

Operadores Lógicos

Símbolo	Exemplo	Equivale
!	!achou !(true) = false !(false) = true	Negação
&&	Op1 && Op2	AND
	Op1 Op2	OR

E lógico (AND)

Op1	Op2	Op1 && Op2
V	V	V
V	F	F
F	V	F
F	F	F

Operadores Lógicos

Símbolo	Exemplo	Equivale
!	!achou !(true) = false !(false) = true	Negação
&&	Op1 && Op2	AND
	Op1 Op2	OR

OU lógico (OR)

Op1	Op2	Op1 Op2
V	V	V
V	F	V
F	V	V
F	F	F

Exemplo

```
x = 2; y = 3; z = 4; achou = false;
```

```
(x < y + 1) && (!achou) && (z == x + 3)
```

Exemplo

```
x = 2; y = 3; z = 4; achou = false;
```

```
(x < y + 1) && (!achou) && (z == x + 3)
```

```
(2 < 3 + 1) && (!false) && (4 == 2 + 3)
```

Exemplo

```
x = 2; y = 3; z = 4; achou = false;
```

```
(x < y + 1) && (!achou) && (z == x + 3)
```

```
(2 < 3 + 1) && (!false) && (4 == 2 + 3)
```

```
(2 < 4) && true && (4 == 5)
```

```
true && true && false
```

```
true && false
```

```
false
```

Funções Matemáticas em C

Incluir a biblioteca math.h

O resultado de cada uma delas é do tipo double

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

#define PI 3.14159265

int main ()
{
    double x, ret, val;

    x = 60.0;
    val = PI / 180.0;
    ret = cos( x*val );
    printf("Cosseno de %lf é %lf graus\n",
x, ret);

    return(0);
}
```

Função	Descrição do comando
floor()	arredonda para baixo
ceil()	arredonda para cima
sqrt()	Calcula raiz quadrada
pow(variável, expoente)	potenciação
sin()	seno
cos()	cosseno
tan()	Tangente
log()	logaritmo natural
log10()	logaritmo base 10

Funções Matemáticas em C

O sorteio de números é muito utilizado.

Pode-se utilizar a hora do sistema como base para geração dos números aleatórios com a função `srand` e `time`, além de definir um intervalo (mínimo e máximo) dos valores que serão sorteados por uma expressão utilizada com a função `rand`.

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

int main () {

    // Inicializa gerador de números
    srand((unsigned) time(NULL));

    // Sorteia número entre 0 e N
    printf("%d\n", (rand() % (N+1)));

    return(0);
}
```