
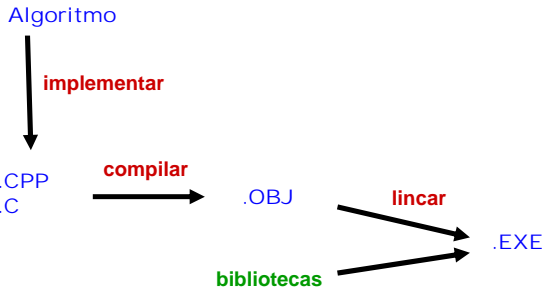
 **Introdução a linguagem C** 1

- Compiladores
- Histórico
- Estrutura básica
- Funções
- Diretiva #include
- Tipos de Dados
- Variáveis
- Constantes
- Comando printf()
- Exercícios
- Anexo 1 - Bases numéricas
- Anexo 2 - Diretiva #define

01 - Introdução www.bracad.com.br


 **Compiladores** 2



```

graph TD
    A[Algoritmo] -- implementar --> B[.CPP  
.C]
    B -- compilar --> C[.OBJ]
    C -- linkar --> D[.EXE]
    E[bibliotecas] --> D
  
```

01 - Introdução www.bracad.com.br

 **Histórico** 3

- *BCPL => B => C*
- 1972 (Lab. Bell) - Dennis Ritchie e Ken Thompson
- *The C Programming Language*
- Brian W. Kernighan e Dennis M. Ritchie
- *Para desenvolvimento de: sistemas operacionais, planilhas eletrônicas, processadores de texto, gerenciadores de banco de dados*
- *Programas em C tendem a ser bastante compactos e de execução rápida*
- *A linguagem C é Case-Sensitive.*

01 - Introdução www.bracad.com.br



Estrutura básica

4

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

void main ( ) {
    clrscr ( );
    printf(" Ling. C ");
    getch ( );
}

#include <stdio.h>
#include <conio.h>

main ( ) {
    clrscr ( );
    printf(" Ling. C ");
    getch ( );
    return(0);
}
```



Funções

5

main () { }
função principal do programa; a execução do programa sempre começa por esta função.

clrscr ();
clear screen - limpa toda a tela.

getch ();
solicita que uma tecla seja pressionada.

return (0);
deve ser colocado sempre que a função main não for do tipo **void**.



Diretiva #include

6

- Inclusão de um programa fonte em outro.

#include <biblioteca.h>
busca em diretório de inclusão \INCLUDE

#include "biblioteca.h"
busca em diretório atual

#include "C:\biblio\biblioteca.h"
busca em diretório específico: \biblio



Tipos de Dados

7

Tipos básicos:

char	8 bits	(caracter(es))
int	16 bits	(inteiro)
float	32 bits	(real)
double	64 bits	(real)
void	0 bits	



Tipos de Dados

8

- Modificadores: short long
signed unsigned
(short e signed não são necessários)

void	--		
char	caracter	1byte	signed: -128 até 127 unsigned: 0 to 255
int	inteiro curto	2bytes	signed: -32768 até 32767 unsigned: 0 to 65535
long int	inteiro longo	4bytes	signed: -2147483648 até 2147483647 unsigned: 0 até 4294967295
float	reais	4bytes	3.4e +/- 38 (6 digitos)
double	reais	8bytes	1.7e +/- 308 (10 digitos)
long double	real longo	10bytes	3.4e-4932 até 1.1e+4932 (10 digitos)



Variáveis

9

- São espaços reservados na memória para armazenamento de valores, que podem ser modificados.
- Todas as variáveis devem ser declaradas;

Exemplos:

```
char resp, letra = 'A';
int a, b = 0, A;
float x = -5.6, y = 6.9, w = 7.58;
double k = 4.5, h;
long int Z = 107089;
unsigned int var = 50000, valor;
```



Constantes

10

- São espaços reservados na memória para armazenamento de valores, que não podem ser modificados.
- Todas as constantes devem ser declaradas;

Exemplos:

```
const char carteira = 'B';
const int zero = 0;
const float pi = 3.14159;
```



printf()

11

Sintaxe

`printf("expr. de controle", lista de argumentos);`

Códigos	Significado
<code>\n</code>	nova linha
<code>\t</code>	tab (tabulação)
<code>\b</code>	retrocesso (backspace)
<code>\"</code>	aspas
<code> </code>	contra barra
<code>\f</code>	salta uma página
<code>\xnn</code>	código ASCII (nn em hexadecimal)



printf()

12

Exemplos

```
printf("Universidade\n");
printf("A letra j \n pronuncia-se jota\n");
printf(" Para mostrar a \\ devo usar \\\\ \n");
printf("\tLinguagem \n\t\t \" C \" ");
printf("Explicação\x80\x83o");
```

```
Universidade
A letra j
pronuncia-se jota.
Para mostrar a \ devo usar \\
Linguagem
" C "
Explicação
```



printf ()

13

<i>Códigos</i>	<i>Formato</i>
<code>%c</code>	<i>caracter</i>
<code>%s</code>	<i>caracteres</i>
<code>%d</code>	<i>inteiro</i>
<code>%u</code>	<i>decimal sem sinal</i>
<code>%f</code>	<i>float</i>
<code>%lf</code>	<i>reais mais longos</i>
<code>%x</code>	<i>hexadecimal</i>
<code>%o</code>	<i>octal</i>
<code>%e</code>	<i>notação científica</i>
<code>%%</code>	<i>porcentagem</i>

01 - Introdução

www.bracad.com.br



printf ()

14

```
#include <stdio.h>
#include <conio.h>
void main ( ) {
    int num = 10;
    float preco = 15.70;
    const double pi = 3.14159;
    clrscr();
    printf ( "\n Numero = %d", num);
    printf ( "\n Preço: R$ %f", preco);
    printf ( "\n PI = %lf", pi);
    getch();
}
```

```
Numero = 10
Preço: R$ 15.70000
PI = 3.14159
```

01 - Introdução

www.bracad.com.br



printf ()

15

```
#include <stdio.h>
void main ( ) {
    int R = 10;
    int valor = 150;
    printf ( "O reajuste foi de %d%%.\n", R);
    printf ( "São %2d alunos.\n", valor);
    printf ( "São %5d alunos.\n", valor);
    printf ( "São %4d alunos.\n", valor);
}
```

```
O reajuste foi de 10%.
São 150 alunos.
São 150 alunos.
São 150 alunos.
```

01 - Introdução

www.bracad.com.br



printf ()

16

```
#include <stdio.h>
void main ( ) {
    float a = 3458.78, b = 4890.21, c = 8;
    printf ( "%4.2f %3.2f \n", a, b);
    printf ( "%7.1f %10.3f\n", a, b);
    printf ( "%.2f %.3f", c, b);
}
```

3458.78	4890.21
3458.8	4890.210
8.00	4890.210



printf ()

17

```
#include <stdio.h>
void main ( ) {
    int num = 21;
    printf("\n%04d", num);
    printf("\n%06d", num);
    printf("\n%6.4d", num);
    printf("\n%6.0d", num);
    printf("\n%c %c %c %c", 'A', 65, 0x41, 0101);
    printf("\nd %x %o", 'A', 'A', 'A');
}
```

0021
000021
0021
21
A A A A
65 41 101



Exercícios

18

1. Com uma única expressão mostre o texto abaixo:

Esta e' a linha "um".
Esta e' a linha "dois".

2. Mostre na tela:

um
dois
três



3. Declare três variáveis inteiras e atribua 1, 2 e 3; três variáveis caracteres e atribua a, b e c; e mostre:

As variáveis inteiras contêm os números: 1, 2 e 3.
As variáveis caracteres contêm os valores: a, b e c.



Exercícios

19

4. Fazer um programa completo para mostrar a tela:

```

/ / / / /
Linguagem
\ \ \ \ \
" C "
/ / / / /

```

5. Identifique os erros no programa:

```

Main {
    print ('Existem /n')
    print ('muitos erros.')
}

```



Exercícios

20

6. Fazer um programa para mostrar a seguinte tela, usando uma variável inteira:

```

:      15
:00000015
:   00015
:15
:0015

```

7. Fazer um programa para mostrar a seguinte tela, usando uma variável real:

```

:16.500
:   16.5
:   16.50

```



Exercícios

21

8. Mostre como os números serão apresentados pela execução do programa:

```

#include <stdio.h>
void main ( ) {
    float a = 987.569;
    int b = 507;
    printf ("%05d\n", b);
    printf ("%3.4f\n", a);
    printf ("%4d\n", b);
    printf ("%6.1f\n", a);
    printf ("%6.3d\n", b);
}

```



Exercícios

22

9. Mostre como os números serão apresentados pela execução do programa:

```
#include <stdio.h>
void main ( ) {
    int x = 15;
    float y = 27.93, z = 51.273;
    printf ("%d\n", x);
    printf ("%8.3f\n", z);
    printf ("%2f\n", y);
    printf ("%5.0d\n", x);
    printf ("%6.1f\n", z);
    printf ("%7.2f\n", y);
}
```


01 - Introdução

www.bracad.com.br



Exercícios

23

10. Mostre como os números serão apresentados pela execução do programa:

```
#include <stdio.h>
void main ( ) {
    int x = 15;
    float y = 27.93, z = 51.273;
    printf ("%6d\n", x);
    printf ("%7.3f\n", y);
    printf ("%2f\n", z);
    printf ("%03d\n", x);
    printf ("%1f\n", y);
    printf ("%7.2f\n", z);
}
```


01 - Introdução

www.bracad.com.br



Exercícios

24

11. Mostre como os números serão apresentados pela execução do programa:

```
#include <stdio.h>
void main ( ) {
    int x = 29;
    float y = 37.48;
    printf ("%8.1f\n", y);
    printf ("%5d\n", x);
    printf ("%3f\n", y);
    printf ("%06d\n", x);
    printf ("%3.4f\n", y);
    printf ("%9.5d\n", x);
}
```


01 - Introdução

www.bracad.com.br



Exercícios

25

12. Complete o programa abaixo com os comandos necessários para apresentar a tela ao lado:

```
#include <stdio.h>
void main () {
    int x = 10;
    float y = 5.4890;
    printf (          );
    printf (          );
    printf (          );
    printf (          );
    printf (          );
}
```

Y = 5.4890
x = 00010
Y = 5.49
x = 11
Y = 5



Anexo 1

26

Dec	Bin	Oct	Hex
00	00000000	000	00
01	00000001	001	01
02	00000010	002	02
03	00000011	003	03
04	00000100	004	04
05	00000101	005	05
06	00000110	006	06
07	00000111	007	07
08	00001000	010	08
09	00001001	011	09
10	00001010	012	0A
11	00001011	013	0B
12	00001100	014	0C
13	00001101	015	0D
14	00001110	016	0E
15	00001111	017	0F
16	00010000	020	10
17	00010001	021	11
18	00010010	022	12
19	00010011	023	13
20	00010100	024	14

Bases numéricas:

- (10) Decimal
- (2) Binária
- (8) Octal
- (16) Hexadecimal



Anexo 1

27

1 byte = 8 bits
 1 Kbyte = 1024 bytes
 1 Mbyte = 1024 Kbytes

1	0	0	1	1	0	1	1
2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
128	64	32	16	8	4	2	1

155₁₀ = 128 + 16 + 8 + 2 + 1
 155₁₀ = 10011011₂

36₁₀ = 32 + 4
 36₁₀ = 00100100₂



Anexo 1

28

Dec	Bin	Oct	Hex
09	000000001001	011	009
16	000000010000	020	010
24	000000011000	030	018
32	000000100000	040	020
56	000100000000	400	100
64	000001000000	100	040
73	000001001001	111	049
273	000100010001	421	111
511	000111111111	777	1FF



Anexo 2 - #define

29

`#define identificador texto`

- Definir **constantes** simbólicas.
- Compilador substitui cada ocorrência do **identificador** no programa por **texto**.
- Identificador escrito em letras maiúsculas.
- Usa-se um comando por linha.
- Não há ponto-e-vírgula após qualquer diretiva.

```
#define PI 3.14159
#define ERRO printf (" \n ERRO. \n")
```



Anexo 2 - Exemplo

30

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

#define PI 3.14
#define ZERO 0
#define STOP printf("Tecle algo para prosseguir.")
#define OP 'S'

void main() {
    clrscr();
    printf("%5.3f\n", PI);
    printf("%d\n", ZERO);
    printf("%c\n", OP);
    STOP;
    getch();
}
```
