

Programação de Computadores I

Aula 3

Daniel Lohmann
Contribuições: Fernando Pacheco

Sumário

- Programação
- Scratch
- Linguagem C e lógica de programação;
- Estrutura de um programa em C
 - Operadores básicos e Variáveis
 - Função *printf* e *scanf*

Programação de Computadores I



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

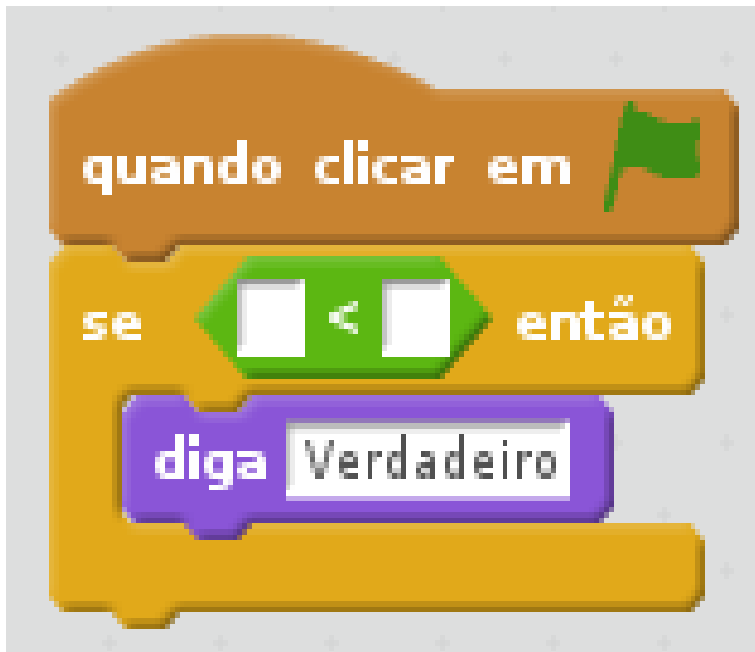
```
int main () {
```

```
    printf(" Hipopotomonstrosesquipedaliofobia ");
```

```
    return (0);
```

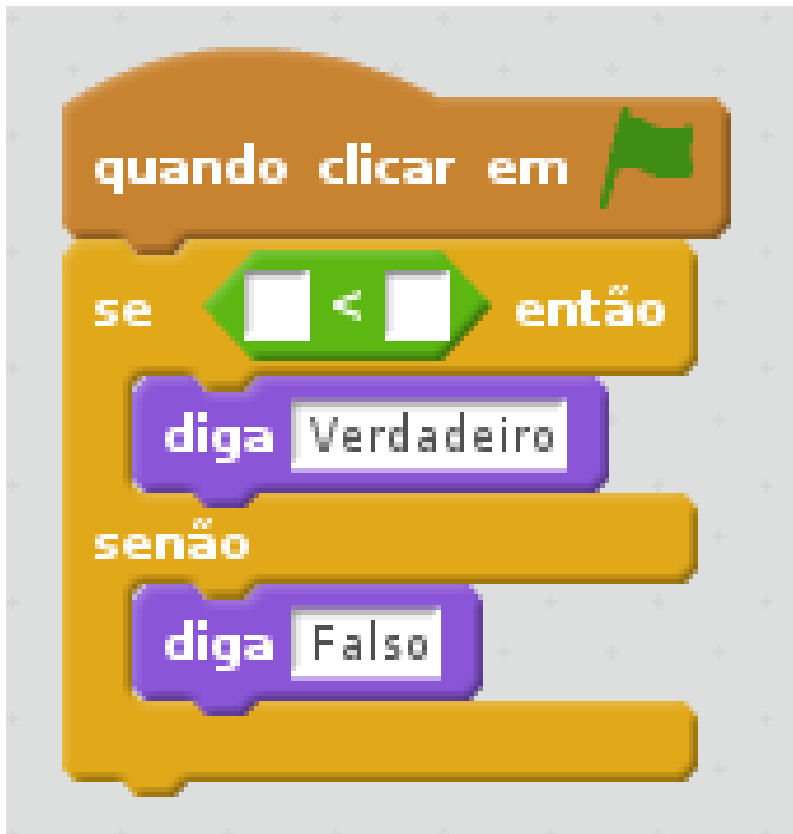
```
}
```

Programação de Computadores I



```
If (condição) {  
    printf("verdadeiro");  
}
```

Programação de Computadores I



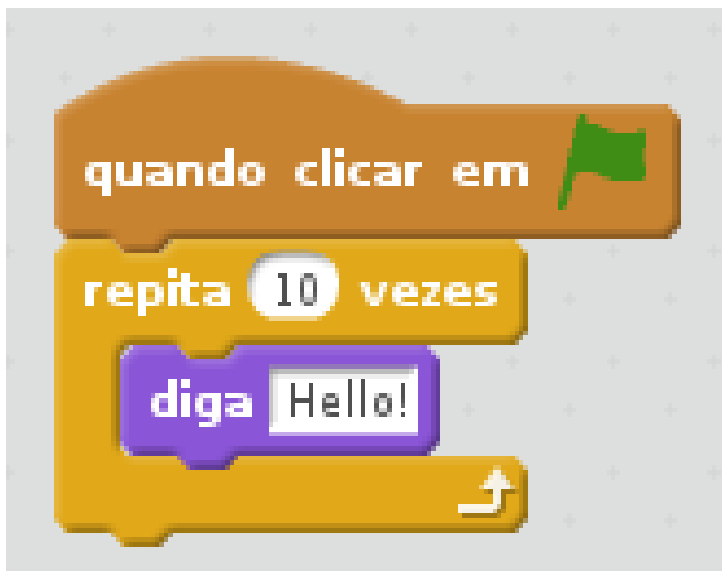
```
If (condição) {  
    printf("Verdadeiro");  
}  
else{  
    printf("Falso");  
}
```

Programação de Computadores I



```
If (condição 1) {  
    printf("Verdadeiro");  
}  
else if (condição 2){  
    printf("Meio Falso");  
}  
else {  
    printf("Falso");  
}
```

Programação de Computadores I



```
For (i = 0 ; i < 10; i++){  
    printf("Hello! ");  
}
```

Programação de Computadores I



```
while (condição_verdadeira )  
{  
    printf("Hello! ");  
}
```


Programação de Computadores I

“condição_verdadeira” em C:

- Booleana TRUE;
- Número diferente de 0(zero);

“condição_falsa” em C:

- Booleana FALSE;
- 0(zero);

Programação de Computadores I

Operadores Relacionais C++	Exemplo de Relação	Significado da Relação
>	$x > y$	x é maior que y
<	$x < y$	x é menor que y
>=	$x \geq y$	x é maior ou igual a y
<=	$x \leq y$	x é menor ou igual a y
==	$x == y$	x é igual a y
!=	$x != y$	x é diferente de y

Programação de Computadores I

Operadores Lógicos

Operador	Ação
&&	And (E)
	Or (Ou)
!	Not (Não)

Programação de Computadores I

Laço com comando **do while** :

- **Executa o laço e depois testa a condição;**
do {
 comando 1;
 comando 2;
 ...
}while(condição);

Programação de Computadores I



Variáveis :

```
int x;
```

```
x = 0;
```

```
x = x+1;
```

Programação de Computadores I

Variáveis : Limites definidos em <limits.h>

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

Variáveis

- Declaração: nome e tipo (e comentário)
`int num_dias; /* número de dias de aula */`
- C é case sensitive (diferencia maiúsculas de minúsculas)
- O que **não** vale como nome de variável em C
 - Começar com número: 1a_nota
 - Usar \$: media\$final
 - Espaço: media final
 - Usar operadores (+, , /, ...): media*final
 - Palavras reservadas: float, int, for...

Variáveis

- Declaração de variáveis para caracteres:
char letras;
Na memória reserva-se 8bits.

Variáveis

- Declaração de variáveis para caracteres:
char letras;
Na memória reserva-se 8bits.

TABELA ASCII

Tabela ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

Variáveis

- Declaração de variáveis para caracteres:

```
char letras;
```

```
Letras = 65;
```

```
letras = 'A';
```

Variáveis

- Declaração de variáveis Ponto flutuante:

```
float num; // norma IEEE 754
```

```
Num = 6.5; // atenção no PONTO
```

Operadores básicos

- Multiplicação: *
- Divisão e também divisão inteira: /
- Adição: +
- Subtração: –
- Resto após divisão inteira: %

- O que faz o programa seguinte?

```
int main()
{
    (3 + 7) * 2;
    return(0);
}
```

Operadores básicos

Operador	Exemplo	Comentário
=	x = y	Atribui o valor de y a x
+=	x += y	Equivale a $x = x + y$
-=	x -= y	Equivale a $x = x - y$
*=	x *= y	Equivale a $x = x * y$
/=	x /= y	Equivale a $x = x / y$
%=	x %= y	Equivale a $x = x \% y$

Estrutura de um programa C

- Comentários: muito importante!
- Dados: “o que vai ser usado”
- Instruções: “como usar”

```
/*  
* Cabeçalho / Comentários  
*/
```

```
//Declaração de dados (globais)
```

```
int main()  
{  
    //Comandos  
    return(0);  
}
```


Estrutura de um programa C

- Comentários
 - Entre `/*` e `*/`
- Em C, linha de código termina com `;`
- Blocos de código são colocados entre chaves `{`
`}`

Função main()

- Primeira função executada em um programa
- Deve retornar um inteiro

```
int main()
```

Programa clássico Hello World

```
#include <stdio.h>

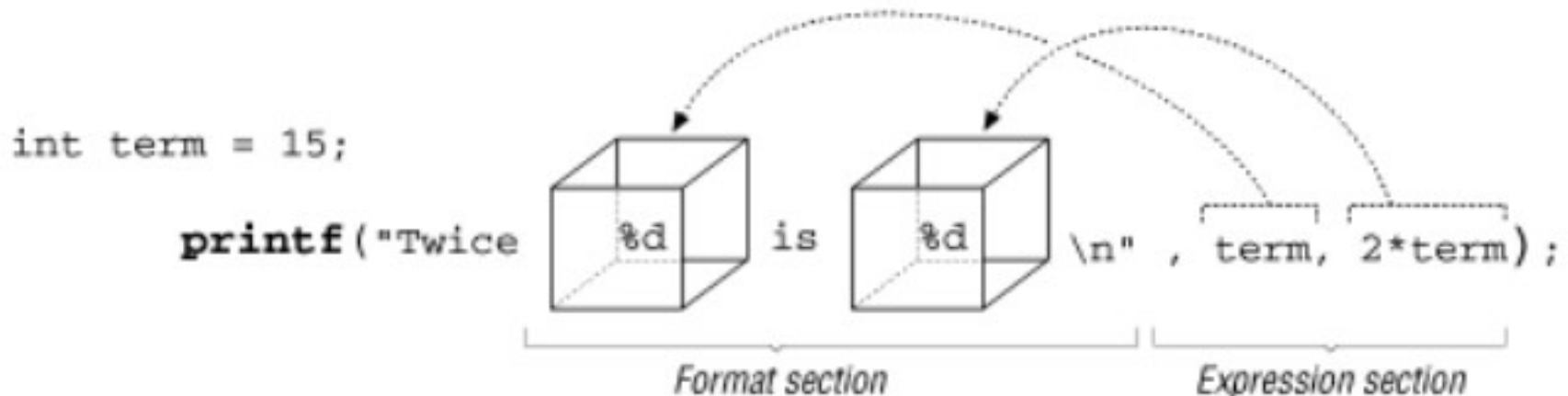
int main()
{
    printf("Hello World!\n");
    return(0);
}
```

Estilo

- Muito importante: COMENTAR o código
 - Comentar é uma arte que pode ser aprendida
 - Programa sem comentários pode se tornar difícil de ler
 - Mas excesso de comentários também é ruim
- Pelo menos, um cabeçalho
- Endentação (facilita leitura para o programador)
 - Para o compilador, é irrelevante
 - Automática nos IDEs
 - Use o sítio <http://indentcode.net>

Função printf

- Para apresentar informação na tela
- Definida em <stdio.h>
 - printf(formato, expressao1, expressao2, ...)
- Mais em <http://www.cppreference.com/wiki/c/io/printf>



Função printf

printf("O valor de %d vezes %d eh %d", a, b, c);



Código	Formato
%c	Um caracter (char)
%d	Um número inteiro decimal (int)
%i	O mesmo que %d
%e	Número em notação científica com o "e"minúsculo
%E	Número em notação científica com o "e"maiúsculo
%f	Ponto flutuante decimal
%g	Escolhe automaticamente o melhor entre %f e %e
%G	Escolhe automaticamente o melhor entre %f e %E
%o	Número octal
%s	String
%u	Decimal "unsigned" (sem sinal)
%x	Hexadecimal com letras minúsculas
%X	Hexadecimal com letras maiúsculas
%%	Imprime um %
%p	Ponteiro

Função printf

Exemplos de formatação da saída printf()

Declaração print()

("%-5.2f", 123.234)

|123.23

saída

|

("%5.2f", 123.234)

| 123.23|

("%10s", "hello")

| hello|

("%-10s", "hello")

|hello|

("%5.7s", "123456789")

| 1234567|

("%010d", 1234)

|0000001234|

Função printf

<u>Sequência de escape</u>	<u>Significado</u>
\a	Caractere Bell (ANSI C)
\b	Caractere de retrocesso (backspace)
\n	Caractere de nova linha
\r	Caractere de retorno de carro
\t	Caractere de tabulação horizontal
\v	Caractere de tabulação vertical (ANSI C)
\\	Caractere de contra-barra
\'	Caractere de aspas simples
\"	Caractere de aspas duplas
\?	Caractere de ponto-de-interrogação
\nnn	Valor ASCII em octal
\xnnn	Valor ASCII em hexadecimal
\0	Caractere nulo

Exercício

- Calcular e apresentar na tela o resultado de $(3 + 7) * 2$

Exercício

- Criar três variáveis: x, y e z
- Atribuir os valores $x=10$, $y=2$ e $z=6$
- Mostrar na tela o resultado das expressões
 - a) $x*y+z$ b) $x*(y+z)$ c) $x\%y$
 - d) $x\%z$ e) $8\%y$ f) $9\%y$
 - g) x/y h) $(x-y)*z-y$
- Cite uma das aplicações das expressões com o % (resto após divisão inteira)?

Exercício (div1.c)

- Qual o resultado do programa seguinte? Está correto? Como corrigir?

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

```
float resposta; // resultado do calculo
```

```
int main()
```

```
{
```

```
    resposta = 1/3;
```

```
    printf("O resultado eh %f\n", resposta);
```

```
    return(0);
```

```
}
```

Exercício (operacoes.c)

- Qual o valor de `var_int` e `var_flut` em cada linha? (Inclua um `printf` após cada operação)

```
int var_int;  
float var_flut;  
  
int main()  
{  
    var_flut = 1.0/2.0;  
    var_int = 1/3;  
    var_flut = (1/2) + (1/2);  
    var_flut = 3.0/2.0;  
    var_int = var_flut;  
    return(0);  
}
```

Exercício

- Qual o resultado do programa seguinte? Está correto? Como corrigir?

```
/******  
 * div.c      *  
*****/  
  
#include <stdio.h>  
float result; /* resultado da divisão */  
int main()  
{  
    result = 7.0 / 22.0;  
    printf("O resultado eh %d\n", result);  
    return (0);  
}
```

Exercício

- Qual o resultado do programa seguinte? Está correto? Como corrigir?

```
/******  
 * doisedois.c  *  
*****/  
  
#include <stdio.h>  
int resposta;  
int main()  
{  
    resposta = 2 + 2;  
    printf("O resultado eh %d\n");  
    return (0);  
}
```

Exercício

- O programa seguinte tem vários erros. Corrija-os.

```
/* Programa com erros */  
#include <Stdio.h>;  
  
int main()  
(  
    float a, b;  
    a = 3.5;  
    printf("Valor de a = %d/n", a);  
    b = 4.0;  
    c = a*a + b;  
    print("Resultado = &f", c);  
    returned 0;  
}
```

Crédito das figuras

- OUALLINE, S. *Practical C Programming*. 3. ed. O'Reilly, 1997.
- Compiler, assembler, linker and loader: a brief story. Disponível em:
<http://www.tenouk.com/ModuleW.html>