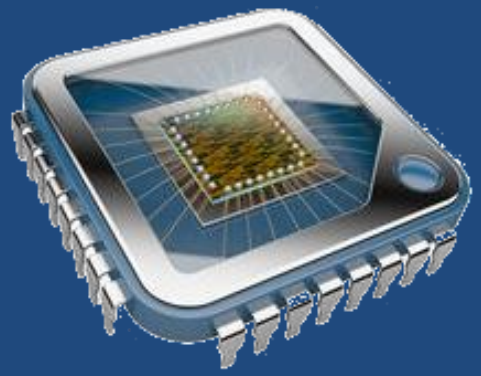
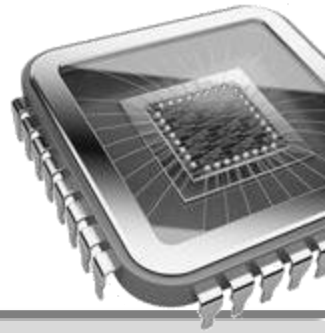


Introdução à Programação
Aula 05

Prof. Max Santana Rolemberg Farias
max.santana@univasf.edu.br
Colegiado de Engenharia de Computação

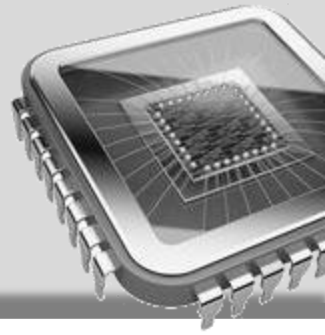
GESE
Grupo de Engenharia de Sistemas Embarcados

CEComp
Engenharia da Computação
UNIVASF



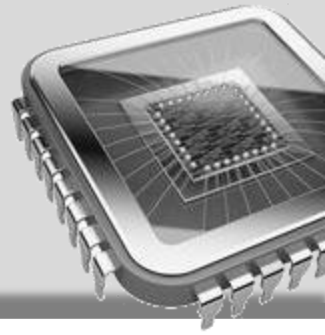
VARIÁVEIS

Variáveis



- São os identificadores fundamentais de qualquer linguagem de programas.
  - Um identificador é uma sequência não nula de caracteres iniciadas com um caractere diferente de dígito
- Uma variável representa um espaço na memória.
  - O espaço de memória ocupado por uma variável pode ser compartilhado por diferentes valores.
    - Uma variável é um espaço de memória que pode conter, a cada tempo, valores diferentes

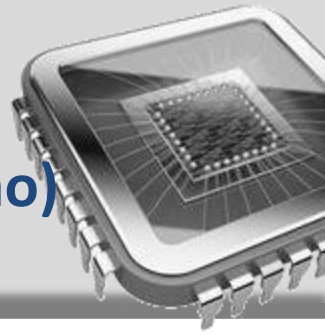
Declaração de Variáveis



- A declaração de uma variável é uma instrução que determina o nome e o tipo com o qual pode ser acessado.
  - As instruções de declaração de variável reserva uma quantidade de memória suficiente para armazenar o tipo especificado.
- Todas as variáveis em C devem ser declaradas antes de ser usada.

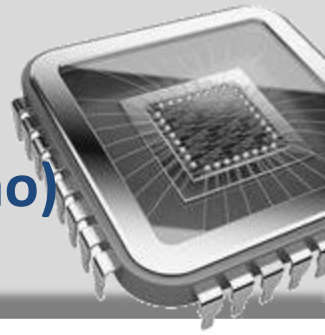
<tipo> <nome_variável>;

Declaração de Variáveis (continuação)



- Escolha nomes significativos para as sua variáveis
  - Ajuda a entender o que o programa faz e previne erros
  - Use quantos caracteres quiser para um nome de variáveis (mas o primeiro é obrigatoriamente uma letra ou sublinhado)
  - O nome de uma variável pode conter letras maiúsculas ou minúsculas, dígitos entre 0 (zero) e 9 (nove) e o caractere de sublinhado

Declaração de Variáveis (continuação)

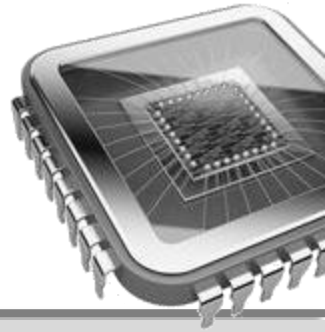


- Para criar mais de uma variável do mesmo tipo em uma instrução, você pode escrever i tipo e, em seguida, os nomes das variáveis, separados por vírgulas

<tipo> variavel1, variavel2, variavel3;

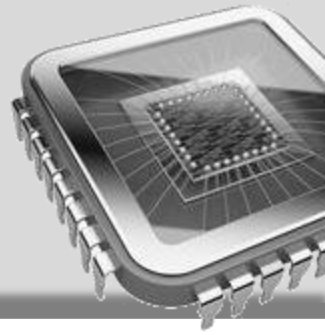
- É possível inicializar uma variável, através de uma opção de inicialização, durante a declaração. Para atribuir um valor a uma variável, você usa o operador =

<tipo> var1 = 0;



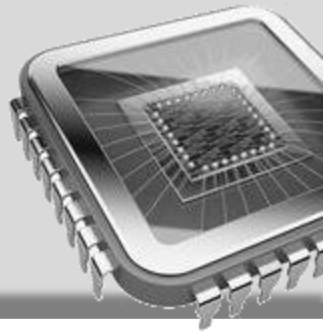
TIPOS DE VARIÁVEIS

Tipos de Variáveis



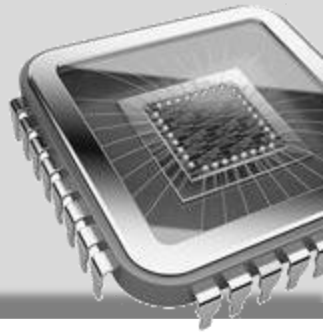
- O tipo de uma variável é necessário para informar ao sistemas operacional a quantidade de memória
- Os tipos básicos da linguagem C são:
  - Inteiros
  - Ponto flutuante
  - Caracteres
- **A linguagem C não tem tipo de dados análogos ao tipo *varient*, onde se podem armazenar todos os tipos básicos possíveis**

Tipos de Variáveis Inteiros



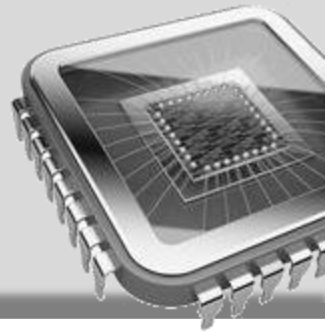
- Os tipos inteiros representam números inteiros
- A linguagem C suporta 10 tipos de inteiros
  - Arquitetura 32 bits e complemento de 2 para negativos
  - Compilador gcc

Tipos de Variáveis Inteiros



Tipo	Bits	Valor mínimo	Valor máximo
signed char	8	-128	127
short int	16	-32.768	32.767
Int	32	-2.147.483.648	2.147.483.647
long int	32	-2.147.483.648	2.147.483.647
long long int	64	-9.223.372.036.854.775.808	9.223.372.036.854.775.807
unsigned char	8	0	255
unsigned short int	16	0	65.535
unsigned int	32	0	4.294.967.295
unsigned long int	32	0	4.294.967.295
unsigned long long int	64	0	18.446.744.073.709.551.615

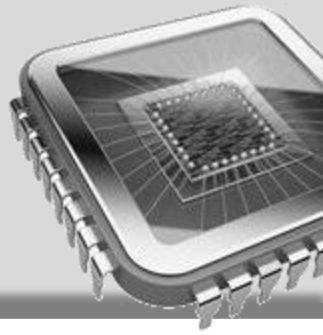
Tipos de Variáveis
Ponto Flutuante



- Os tipos reais de ponto flutuante são implementados com a representação sinal, mantissa e expoente.

Tipo	Sinal	Expoente	Mantissa
float	1	8	23
double	1	11	52
long double	1	64	63

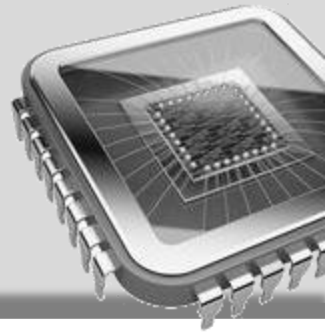
Tipos de Variáveis
Ponto Flutuante



Tipo	Bits	Valor mínimo	Valor máximo
float	32	$1{,}17549 \times 10^{-38}$	$3{,}40282 \times 10^{38}$
double	64	$2{,}22507 \times 10^{-308}$	$1{,}79769 \times 10^{308}$
long double	128	$3{,}3621 \times 10^{-4932}$	$1{,}18973 \times 10^{4932}$

```
...
  printf("float: %d bytes \n", sizeof( float));
  printf("double: %d bytes \n", sizeof(double));
  printf("long double: %d bytes \n", sizeof(long double));
...
```

Tipos de Variáveis
Ponto Flutuante

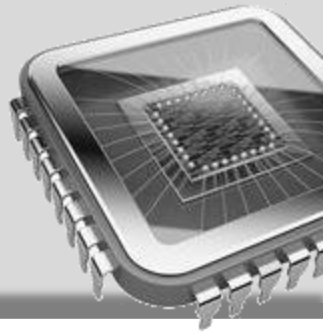


- Os valores positivos mínimos e máximos para cada tipo são definidos em float.h

Tipo	Menor Valor	Maior Valor
float	FLT_MIN	FLT_MAX
double	DBL_MIN	DBL_MAX
long double	LDBL_MIN	LDBL_MAX

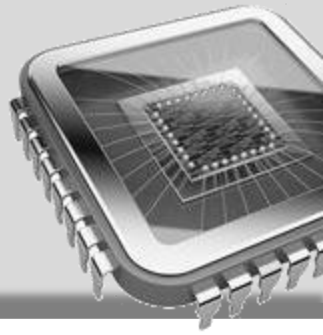
```
#include<float.h>
...
  printf (“float: %g a %g \n", FLT_MIN, FLT_MAX);
  printf(“double: %g a %g \n", DBL_MIN, DBL_MAX);
  printf(“long double: %Lg a %Lg \n", LDBL_MIN, LDBL_MAX);
...
```

Tipos de Variáveis
Caracteres



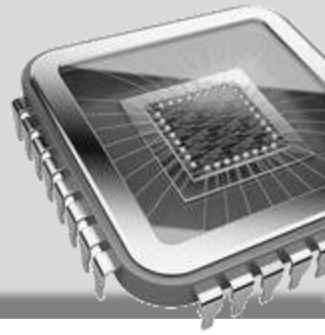
- Em C os caracteres são representados nos tipos char.
  - Os caracteres são representados por valores numéricos segundo uma codificação padrão definida.
  - O conjunto de caracteres podem ser estendidos, mas, em sua forma básica, totalizam 128 caracteres
    - Incluindo as letras maiúscula e minúsculas, os dígitos e os caracteres gráficos.

Tipos de Variáveis
Caracteres (Padrão ASCII)



- O padrão ASCII é um padrão comumente adotado para a representação binária de caracteres.
  - Cada caractere é codificado em 7 bits, possibilitando a representação de 128 símbolos.

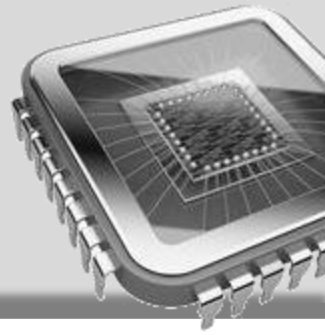
Tipos de Variáveis
Caracteres (ASCII: Normal)



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	&#127;	DEL

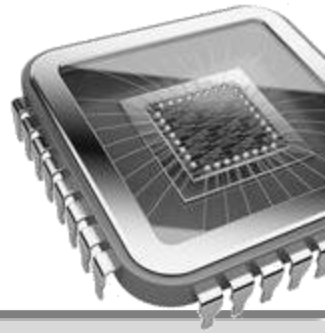
Source: www.LookupTables.com

Tipos de Variáveis
Caracteres (ASCII: Estendida)



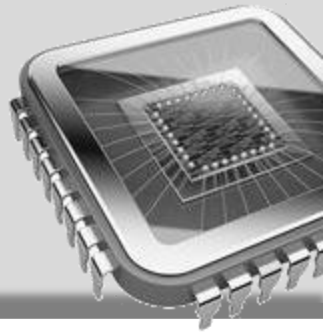
128	Ç	144	É	160	á	176	▒	192	└	208	╨	224	α	240	≡
129	ü	145	æ	161	í	177	▒	193	┴	209	╤	225	ß	241	±
130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
131	â	147	ô	163	ú	179	│	195	├	211	╙	227	π	243	≤
132	ä	148	ö	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷
135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙
138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
139	ï	155	¢	171	½	187	╗	203	╦	219	█	235	δ	251	√
140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	n
141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²
142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	

Source: www.LookupTables.com



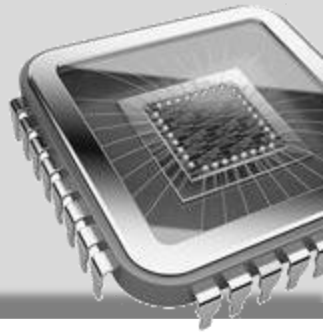
BIBLIOTECA PADRÃO

Biblioteca Padrão

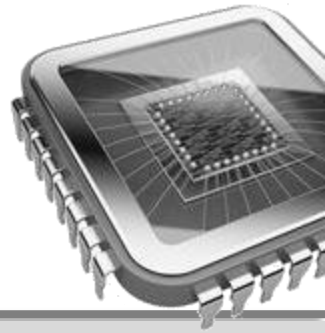


- A linguagem C disponibiliza um conjunto de bibliotecas (conjunto de funções) para serem incorporadas a um programa C.
- As funções dessas bibliotecas são declaradas em um arquivo-cabeçalho.
  - Os arquivos-cabeçalho são organizados por finalidade ou área de aplicação.

Biblioteca Padrão (continuação)

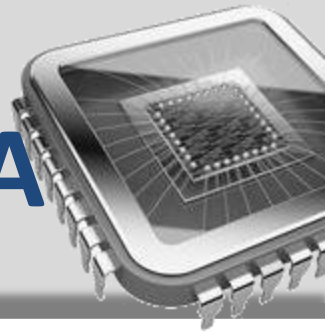


- Os arquivos-cabeçalhos das bibliotecas são incluídos ao programa através da diretiva de pré-processamento *#include*
  - Para incluir arquivos-cabeçalhos do sistema (criados durante a instalação do compilador) as diretivas são referenciadas entre chaves angulares (< >).
  - Os arquivos-cabeçalhos do usuário (desenvolvidos pelo programador) são referenciados através de aspas duplas (“ “).



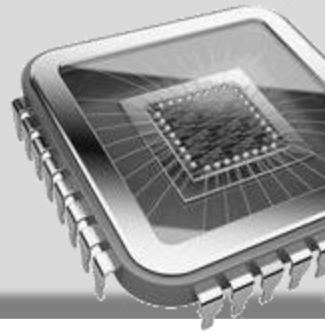
FUNÇÕES DE ENTRADA E SAÍDA

FUNÇÕES DE ENTRADA E SAÍDA



- A linguagem C disponibiliza uma biblioteca padrão de entrada e saída chamada **stdio.h**
  - Std = standard (padrão)
  - Io = input/output (entrada/saída)
  - As funções da biblioteca **stdio.h** estão associadas a entrada e saída padrão do sistema computacional.
    - Teclado
    - Monitor

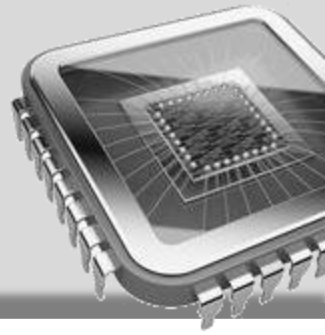
Função de saída



- A função de saída do **stdio.h** é chamada de **printf**
  - A função **printf** imprime mensagens e valores diretamente na saída padrão
  - As mensagens impressas devem ser especificadas entre aspas duplas (sequência de caracteres)

printf(<sequencia_caracteres>);

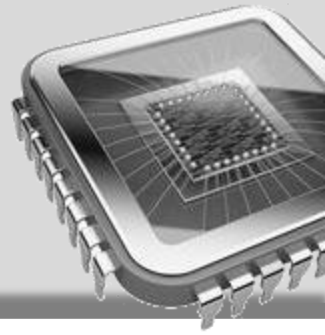
Função de saída (continuação)



- Quando existe a necessidade de imprimir valores de um tipo básico, a biblioteca **stdio.h** oferece as diretivas de impressão para formatar tipos básicos em uma sequência de caracteres.
  - As diretivas de impressão são incluídas na sequência de caracteres através do caractere %

printf(“<sequência_caractere>[%[<tamanho>.<precisão>]<especificação_formato>]”, [lista_valores])

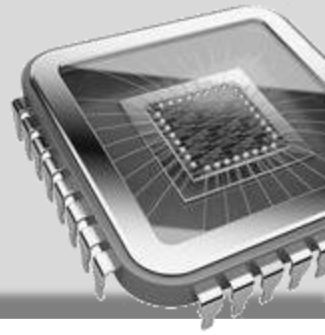
Função de saída (continuação)



- Especificadores de formato (inteiro)
  - Usadas para imprimir valores inteiros

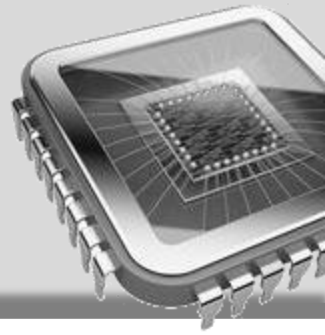
Especificador	Descrição
%d, %i	O argumento correspondente deve ser do tipo inteiro. E o valor será impresso como um inteiro no formato decimal.
%O	O argumento deve ser do tipo inteiro sem sinal. O valor será impresso como um inteiro não sinalizado no formato octal.
%U	O argumento deve ser do tipo inteiro sem sinal. O valor será impresso como um inteiro não sinalizado no formato decimal.
%x, %X	O argumento deve ser do tipo inteiro sem sinal. O valor será impresso como um inteiro não sinalizado no formato hexadecimal.

Função de saída (continuação)



```c
#include <stdio.h>

int main () {
  printf("%d %i %o %u %x %X\n", 155, 155, 155, 155, 155, 155);
  printf("%d %i %o %u %x %X\n", -155, -155, -155, -155, -155, -155);
  return 0;
}
```

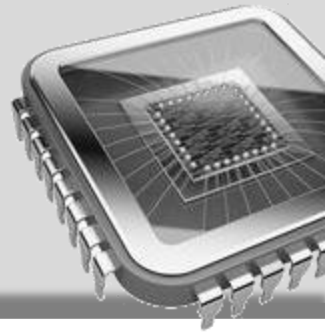
Função de saída (continuação)



- Especificadores de formato (ponto flutuante)
  - Usadas para imprimir valores reais

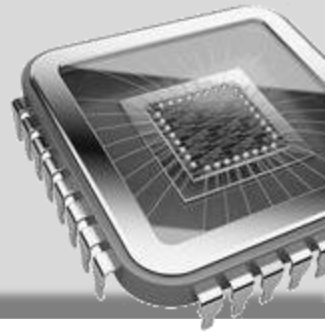
Especificador	Descrição
%f, %F	O argumrnto deve ser do tipo real de ponto flutuante. E o valor será impresso como um número real de ponto flutuante no formato decimal.
%e, %E	O argumento deve ser do tipo real de ponto flutuante. O valor será impresso como um real de ponto flutuante no formato científico.
%g, %G	O argumento deve ser do tipo real de ponto flutuante. O valor será impresso como um real de ponto flutuante no formato da diretiva F ou E, dependendo da precisão.
%a, %A	O argumento deve ser do tipo real de ponto flutuante. P valor será impresso no formato hexadecimal.

Função de saída (continuação)



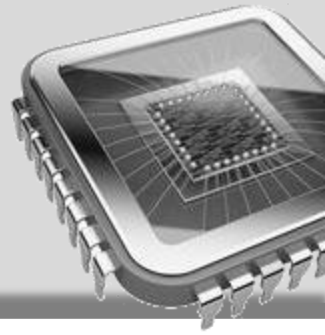
```c
#include <stdio.h>

int main () {
   printf("%f %F\n", 258.97, 258.9701238);
   printf("%e %E\n", 258.97, 258.9701238);
   printf("%g %G\n", 258.97, 258.9701238);
   printf("%g %G\n", 0.000125, 0.000125);
   printf("%g %G\n", 0.0000125, 0.0000125);
   printf("%a %A\n", 258.97, 258.9701238);
   return 0;
}
```

Função de saída (continuação)



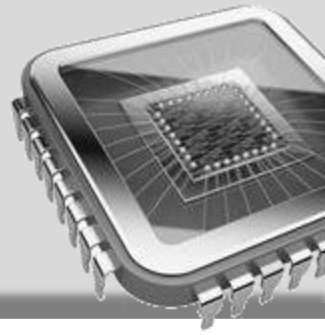
- Nas diretivas %f e %e, os valores são arredondados, sendo a quantidade de dígitos da parte fracionária determinada pela precisão (6 dígitos sem precisão).
- Na diretiva %g os zeros da parte fracionária são removidos.

Função de saída (continuação)



```c
#include <stdio.h>

int main () {
   printf("%3.3f %3.3F\n", 258.97, 258.9701238);
   printf("%3.3e %3.3E\n", 258.97, 258.9701238);
   printf("%3.3g %3.3G\n", 258.97, 258.9701238);
   printf("%3.3g %3.3G\n", 0.000125, 0.000125);
   printf("%3.3g %3.3G\n", 0.0000125, 0.0000125);
   printf("%3.3a %3.3A\n", 258.97, 258.9701238);
   return 0;
}
```

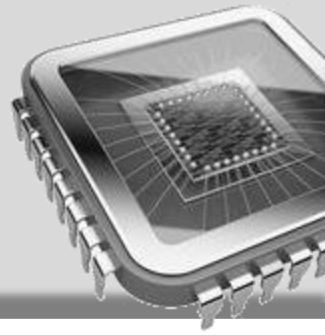
Função de saída (continuação)



- Especificadores de formato (caractere)
  - Usadas para imprimir valores reais

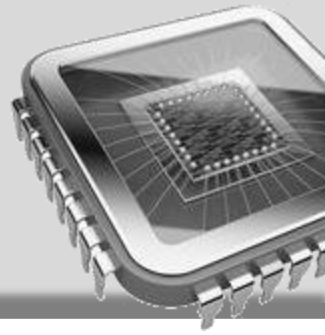
Especificador	Descrição
%c	O argumento correspondente deve ser do tipo inteiro. E o valor é convertido no tipo caractere sem sinal e imprime o caractere
%s	O argumento deve ser um ponteiro do tipo caractere que aponta para uma cadeia de caracteres. E apenas a quantidade de caracteres especificada pela precisão é impressa, e se a precisão não é fornecida é impressa toda a cadeia.

Função de saída (continuação)



```c
#include <stdio.h>

int main () {

  printf("%s d%c %s %c\n", "Fundamentos ", 'e', "programação usando ", 'C');

  return 0;
}
```

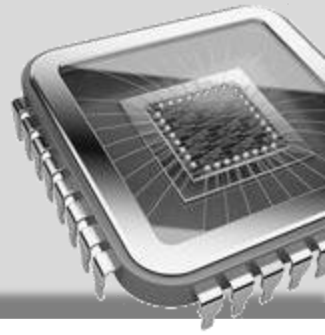
Função de saída (continuação)



- Especificadores de formato (caractere %)
  - Usadas para imprimir o caractere %

Especificador	Descrição
%%	Essa diretiva não possui argumentos correspondente. O caractere % é impressp

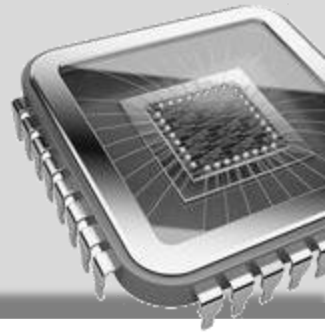
Função de Saída (continuação)



- Constantes de barra invertida
  - Tabela simplificada

Especificador	Descrição
\n	Nova linha
\”	Aspas
\’	Apóstofro
\0	Nulo (zero decimal)
\\	Barra invertida
\t	Tabulação
\a	Alerta (beep)
\b	Retorno do cursor

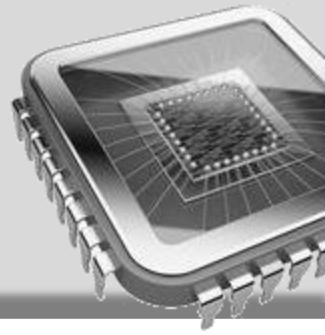
Função de Entrada



- A função de entrada do **stdio.h** é chamada de **scanf**
  - A função **scanf** é o modo mais simples de ler um valor do teclado e armazená-lo em uma variável

scanf(<especificador_formato>, &<variável>);

Função de entrada (continuação)



- Especificadores de formato
  - Tabela simplificada de especificadores de formato

Especificador	Descrição
%c	Um caractere, para armazenar o valor em uma variável do tipo char
%d	Um número inteiro decimal, para armazenar o valor em uma variável do tipo int
%f	Um número de ponto flutuante decimal, para armazenar o valor em uma variável do tipo ponto flutuante.
%s	Uma string, para armazenar em variáveis que possam armazenar strings.