



# Linguagem de Programação

## Comandos de Entrada e Saída

*printf() / scanf() / getchar()*

**Gil Eduardo de Andrade**





# Comandos de Entrada e Saída

## Introdução

- Os comandos de entrada e saída permitem ao programador interagir com o usuário;
- Os comandos de saída permitem que informações sejam enviadas ao usuário;
- Os comandos de entrada permitem que informações especificadas pelo usuário sejam recebidas pelo programa codificado;





# Comandos de Entrada e Saída

## Comando de Saída

- O principal, e mais utilizado comando de saída, na linguagem C, é o ***printf()*** (visto na aula anterior);
- O ***printf()*** permite o envio de textos estáticos e também o valor de variáveis ao monitor do computador (saída padrão);
  - ***Ex.: printf(“Gil Eduardo de Andrade”);***





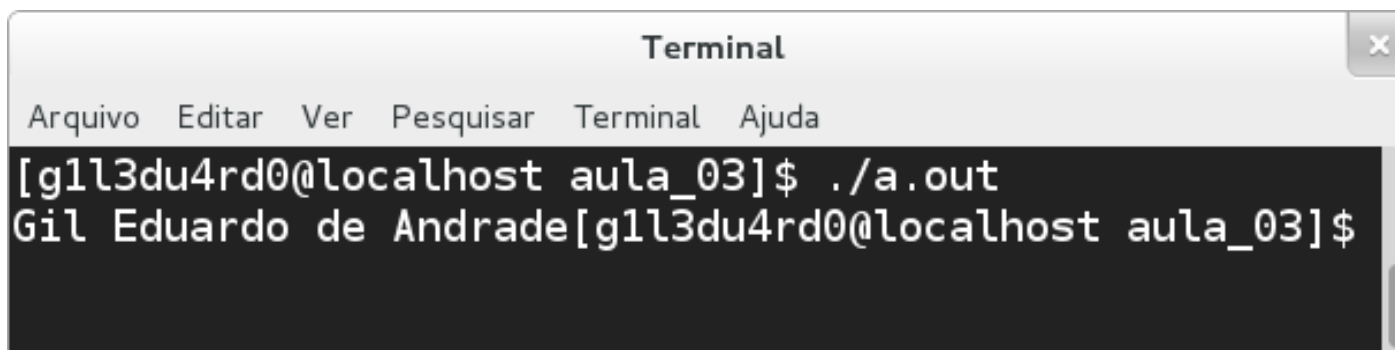
# Comandos de Entrada e Saída

## Sintaxe do comando *printf()*:

```
#include <stdio.h>

int main() {

    printf("Gil Eduardo de Andrade");
    return 0;
}
```



```
Terminal
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@localhost aula_03]$ ./a.out
Gil Eduardo de Andrade[g1l3du4rd0@localhost aula_03]$
```





# Comandos de Entrada e Saída

## Caracteres especiais para o comando *printf()*:

- “\n”: o ‘barra-n’ permite pular linhas momento em que utilizamos o comando *printf()*;
  - *Ex.: printf(“Gil\nEduardo\nAndrade\n”);*
- “\t” o ‘barra-n’ permite que tabulações sejam criadas quando utilizamos o comando *printf()*;
  - *Ex.: printf(“\tGil\n\t\tEduardo\n\t\t\tAndrade\n”);*



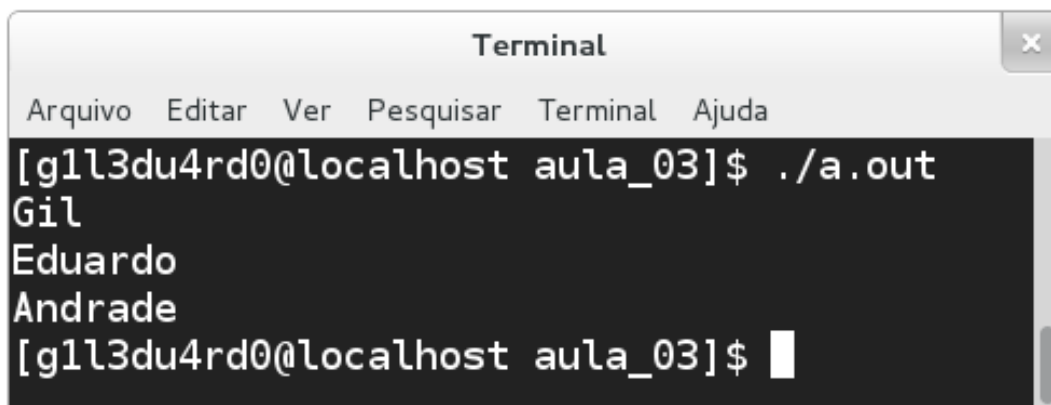
# Comandos de Entrada e Saída

## Caracteres especiais – `printf("\n")`:

```
#include <stdio.h>

int main() {

    printf("Gil\nEduardo\nAndrade\n");
    return 0;
}
```



```
Terminal
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@localhost aula_03]$ ./a.out
Gil
Eduardo
Andrade
[g1l3du4rd0@localhost aula_03]$
```



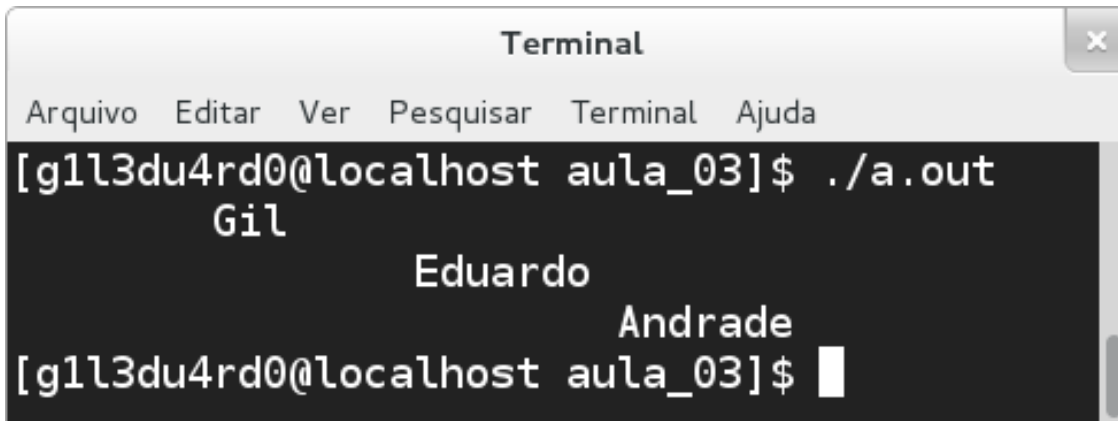
# Comandos de Entrada e Saída

## Caracteres especiais – `printf("\t")`:

```
#include <stdio.h>

int main() {

    printf("\tGil\n\t\tEduardo\n\t\t\tAndrade\n");
    return 0;
}
```



```
Terminal
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@localhost aula_03]$ ./a.out
    Gil
          Eduardo
                Andrade
[g1l3du4rd0@localhost aula_03]$
```





# Comandos de Entrada e Saída

## *printf() para exibir o conteúdo de variáveis:*

- “%i ou %d”: o ‘percentual-i ou d’ permite exibir o conteúdo de variáveis inteiras (*int*);
  - *Ex.: printf(“%i”, nomeA);*
- “%c” o ‘percentual-c’ permite exibir o conteúdo de variáveis caractere (*char*);
  - *Ex.: printf(“%c”, nomeC);*







# Comandos de Entrada e Saída

## *printf()* para exibir o conteúdo de variáveis:

- “%f”: o ‘percentual-f’ permite exibir o conteúdo de variáveis ponto flutuante (*float*);
  - *Ex.: printf(“%f”, nomeD);*
- Observe (em todos os exemplo) que após as aspas duplas, é colocada uma vírgula e logo após esta o nome da variável que deseja-se exibir o conteúdo;





# Comandos de Entrada e Saída

*printf()* para exibir o conteúdo de variáveis:

```
#include <stdio.h>

int main() {

    int nomeA, nomeB;
    char nomeC;
    float nomeD, nomeE;

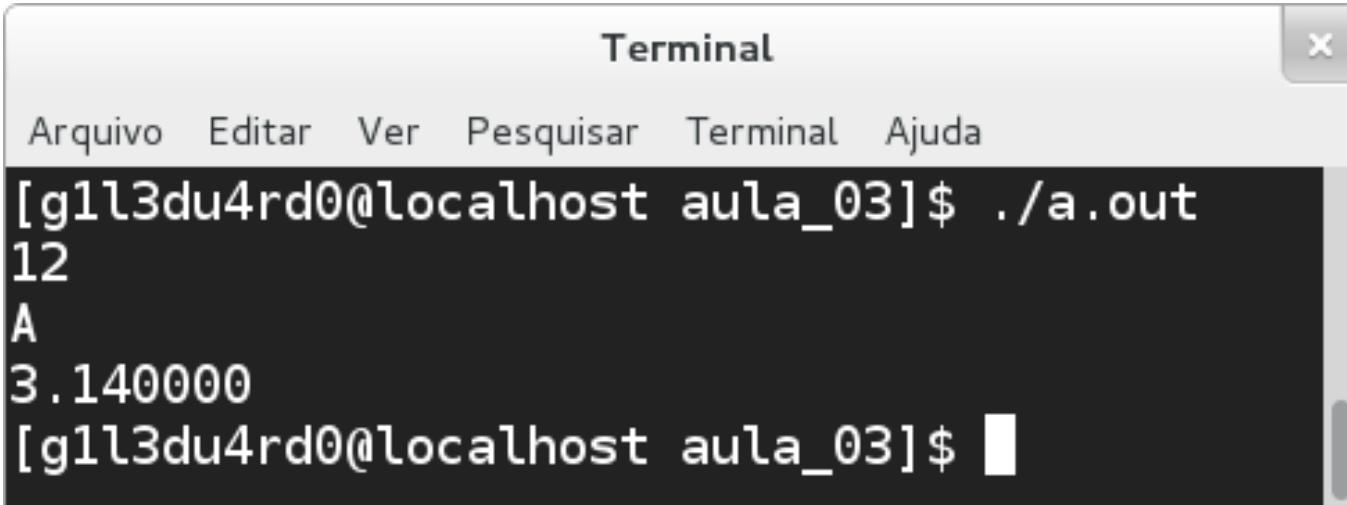
    nomeA = 12;
    nomeC = 'A';
    nomeD = 3.14;

    printf("%i\n%c\n%f\n", nomeA, nomeC, nomeD);
    return 0;
}
```



# Comandos de Entrada e Saída

*printf()* para exibir o conteúdo de variáveis:



```
Terminal
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@localhost aula_03]$ ./a.out
12
A
3.140000
[g1l3du4rd0@localhost aula_03]$
```



# Comandos de Entrada e Saída

*printf()* para exibir o conteúdo de variáveis:

- “%.2f”: o ‘percentual-f’ quando utilizado com um ponto e um valor inteiro permite especificar a quantidade casas depois da vírgula que devem ser apresentadas;
  - *Ex.: printf(“%.2f”, nomeD);*





# Comandos de Entrada e Saída

*printf() para exibir o conteúdo de variáveis:*

```
#include <stdio.h>

int main() {

    int nomeA, nomeB;
    char nomeC;
    float nomeD, nomeE;

    nomeA = 12;
    nomeC = 'A';
    nomeD = 3.14;

    printf("%i\n%c\n%.2f\n", nomeA, nomeC, nomeD);
    return 0;
}
```



# Comandos de Entrada e Saída

*printf()* para exibir o conteúdo de variáveis:

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/C... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ ./a.out
12
A
3.14
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$
```





# Comandos de Entrada e Saída

## Comandos de Entrada:

- O principal, e mais utilizado comando de entrada, na linguagem C, é o *scanf()*;
- O *scanf()* permite ao programador ler informações fornecidas pelo usuário em tempo de execução, via teclado (entrada padrão);





# Comandos de Entrada e Saída

## Comandos de Entrada:

- A sintaxe do comando *scanf()* é:
  - *scanf(“%tipo\_variavel”, &nome\_varivel);*
- Onde *tipo\_variavel* indica o tipo do valor de entrada que deseja-se obter;







# Comandos de Entrada e Saída

## Comandos de Entrada:

- “%i”: para obter um valor *inteiro*;
- “%c”: para obter um *caractere*;
- “%f”: para obter um valor *float*;
- “%d”: para obter um valor *decimal*;



# Comandos de Entrada e Saída

## Exemplos `scanf("%i")`:

```
#include <stdio.h>

int main() {

    int inteiro;
    char letra;
    float flutuante;

    printf("Digite um valor inteiro: ");
    __fpurge(stdin);
    scanf("%i", &inteiro);

    printf("Valor: %i\n", inteiro);

    return 0;
}
```

Destaque para o comando `__fpurge(stdin)`, que deve ser utilizado quando efetuamos a leitura de um caractere, ele permite atualizar (limpar) o buffer de entrada, garantindo que outros comandos `scanf()` que vierem na sequência do código-fonte não sejam ignorados. No sistema operacional Windows deve-se substituí-lo pelo comando `fflush(stdin)`.



# Comandos de Entrada e Saída

## Exemplos *scanf*("%i"):

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compa... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ ./a.out
Digite um valor inteiro: 12
Valor: 12
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ █
```





# Comandos de Entrada e Saída

## Exemplos `scanf("%c")`:

```
#include <stdio.h>

int main() {

    int inteiro;
    char letra;
    float flutuante;

    printf("Digite um caractere: ");
    __fpurge(stdin);
    scanf("%c", &letra);

    printf("Letra: %c\n", letra);

    return 0;
}
```



# Comandos de Entrada e Saída

## Exemplos *scanf*("%c"):

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compa... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ ./a.out
Digite um caractere: G
Letra: G
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ █
```





# Comandos de Entrada e Saída

## Exemplos `scanf("%f")`:

```
#include <stdio.h>

int main() {

    int inteiro;
    char letra;
    float flutuante;

    printf("Digite um valor flutuante: ");
    __fpurge(stdin);
    scanf("%f", &flutuante);

    printf("Valor: %.2f\n", flutuante);

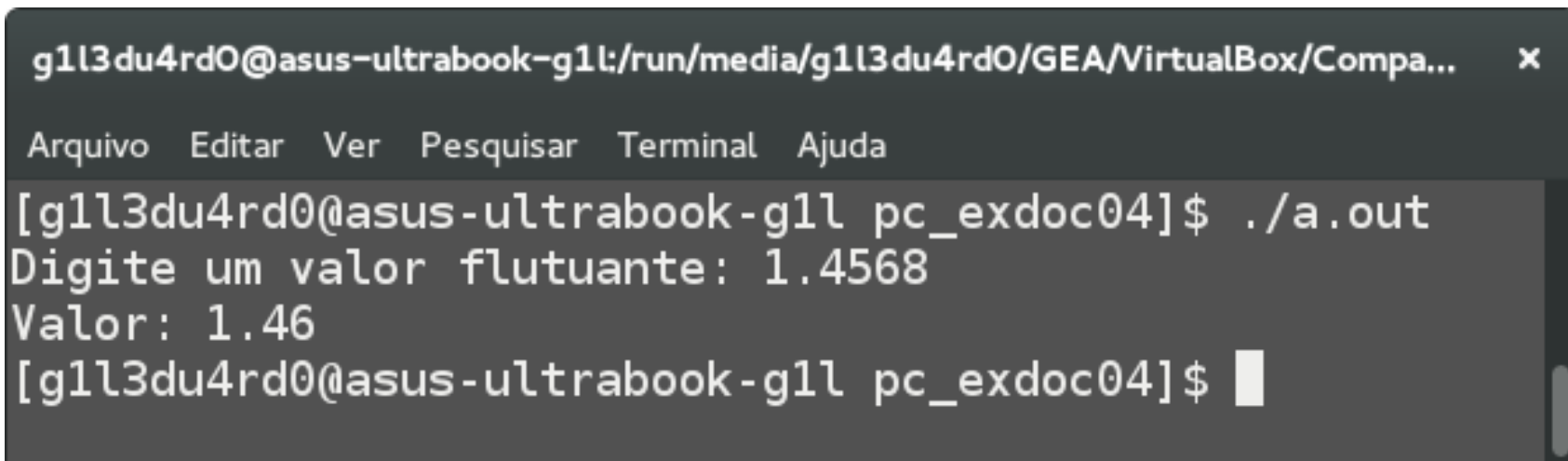
    return 0;
}
```





# Comandos de Entrada e Saída

## Exemplos *scanf*("%f"):



```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compa... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ ./a.out
Digite um valor flutuante: 1.4568
Valor: 1.46
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$
```





# Comandos de Entrada e Saída

## Lendo vários valores com apenas um *scanf()*:

- É possível efetuar a leitura de vários valores através da utilização de apenas um comando *scanf()*;
- Os múltiplos valores de entrada podem ou não ser do mesmo tipo;
  - **Ex.:** *scanf(“%i %c %f”, &inteiro, &letra, &flutuante);*





# Comandos de Entrada e Saída

## Lendo vários valores com apenas um *scanf()*:

```
#include <stdio.h>

int main() {

    int inteiro;
    char letra;
    float flutuante;

    printf("Digite um (int) (char) (float): ");
    __fpurge(stdin);
    scanf("%i %c %f", &inteiro, &letra, &flutuante);

    printf("Valores: %i %c %.2f\n", inteiro, letra, flutuante);

    return 0;
}
```

Observe que os parâmetros do *scanf()* - *%i %c %f* são, obrigatoriamente, separados por um espaço. E as variáveis que receberão os valores digitados na mesma ordem dos parâmetros especificados.



# Comandos de Entrada e Saída

## Comando *getchar()*:

- O comando *getchar()* permite efetuar a leitura de caracteres, de forma análoga ao comando *scanf("%c");*
- Entretanto sua sintaxe é mais simples, pois é preciso apenas atribuí-lo a variável que receberá o caractere digitado;
  - ***Ex.:*** *letra = getchar();*





# Comandos de Entrada e Saída

## Comando *getchar()*:

```
#include <stdio.h>

int main() {

    char letra;

    printf("Digite um caractere: ");
    __fpurge(stdin);
    letra = getchar();
    printf("Letra: %c\n", letra);

    return 0;
}
```

O comando *getchar()*, pode, e normalmente é utilizado, quando há a necessidade do programa parar para apresentar informações ao usuário, aguardando o mesmo pressionar a tecla ENTER, como mostra o próxima slide.





# Comandos de Entrada e Saída

## Comando *getchar()*:

```
#include <stdio.h>

int main() {

    char letra;

    printf("SALDO BANCÁRIO: R$ 2.000,00\n\n");
    printf("Pressione ENTER para voltar ao menu principal...");
    __fpurge(stdin);
    getchar();

    return 0;
}
```



# Comandos de Entrada e Saída

## Comando *getchar()*:

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compa... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ ./a.out
Digite um caractere: X
Letra: X
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ █
```

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Compa... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ ./a.out
SALDO BANCÁRIO: R$ 2.000,00
Pressione ENTER para voltar ao menu principal... █
```



# Linguagem de Programação

Código ASCII e *Casting* de Tipo

Gil Eduardo de Andrade





# Código ASCII e *Casting* de Tipo

## Código ASCII:

- O ASCII é um código numérico padrão, utilizado para representar os caracteres, entendido por quase todos os computadores;
- O ASCII utiliza uma escala decimal que vai do número 0 - 127, e pode ser apresentado no formato de tabela;



# Tabela ASCII

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

Source: [www.LookupTables.com](http://www.LookupTables.com)

Fonte: <http://www.ebah.com.br/content/ABAAfv5sAA/tabela-ascii>



# Código ASCII e *Casting* de Tipo

## Exemplo – Código ASCII:

```
#include <stdio.h>

int main() {

    int valor = 65;
    char caractere = 'X';

    printf("> 'int' PARA 'char': %c", valor);
    printf("\n> 'char' PARA 'int': %i\n", caractere);

    return 0;
}
```

Repare que a variável “valor” do tipo “int” é mostrada pelo printf() como um variável “char”, assim como a variável “caractere” do tipo “char” é mostrada como uma variável “int” pelo segundo printf().

# Código ASCII e *Casting* de Tipo

## Exemplo – Código ASCII:

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Com... x
Arquivo Editar Ver Pesquisar Terminal Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ ./a.out
> 'int' PARA 'char': A
> 'char' PARA 'int': 88
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$
```



Pela execução do código apresentado no slide anterior é possível perceber que: o valor decimal “65” armazenado na variável “valor” corresponde ao código *ascii* da letra “A”, assim como a letra “X” possui como código *ascii* o valor decimal “88”.





# Código ASCII e *Casting* de Tipo

## *Casting* de Tipos:

- O *casting* de tipos nos permite converter tipos de variáveis, como por exemplo: variáveis *int* em *char*, e vice-versa;
- Basicamente existem dois tipos de *casting* na linguagem C: implícito e explícito;





# Código ASCII e *Casting* de Tipo

## *Casting* de Tipos:

- ***Casting Implícito:*** é utilizado para conversões de tipos mais simples, onde não há a necessidade de especificar o tipo da conversão;
  - **Ex.: *char* c = 78;**





# Código ASCII e *Casting* de Tipo

## *Casting* de Tipos:

- ***Casting Explícito:*** é utilizado para conversões de tipos mais complexas, onde há a necessidade de especificar o tipo da conversão;
  - **Ex.:** *float* media = (*float*) 10 / 4;



# Código ASCII e *Casting* de Tipo

## Exemplo – *Casting* de Tipos:

```
#include <stdio.h>

int main() {

    char c = 78;
    float m = 10/4;
    float media = (float) 10/4;

    printf("> Variável c = %c", c);
    printf("\n> Variável m = %.1f", m);
    printf("\n> Variável media: %.1f\n", media);

    return 0;
}
```

Observe que o cálculo “10/4” está sendo efetuado duas vezes, utilizando um *casting* “implícito” e outro “explícito”, armazenado nas variáveis “m” e “media” respectivamente.

# Código ASCII e *Casting* de Tipo

## Exemplo – *Casting* de Tipos:

```
g1l3du4rd0@asus-ultrabook-g1l:/run/media/g1l3du4rd0/GEA/VirtualBox/Com... x
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$ ./a.out
> Variável c = N
> Variável m = 2.0
> Variável media: 2.5
[g1l3du4rd0@asus-ultrabook-g1l pc_exdoc04]$
```



É possível observar pela execução do exemplo que se o *casting* explícito não for utilizado para efetuar o cálculo (10/4) o resultado da operação não é convertida para *float*. Sendo assim a variável  $m=2.0$ , enquanto a variável  $media=2.5$



# Comandos de Entrada e Saída

## Exemplos Utilizados no Documento

[http://www.gileduardo.com.br/ifpr/pci/downloads/pc\\_exdoc04.zip](http://www.gileduardo.com.br/ifpr/pci/downloads/pc_exdoc04.zip)

## Mais Exemplos sobre o Conteúdo

[http://www.gileduardo.com.br/ifpr/pci/downloads/pc\\_ex04.zip](http://www.gileduardo.com.br/ifpr/pci/downloads/pc_ex04.zip)

## Exercícios sobre o Conteúdo

[http://www.gileduardo.com.br/ifpr/pci/downloads/pc\\_pratica04.pdf](http://www.gileduardo.com.br/ifpr/pci/downloads/pc_pratica04.pdf)