



INSTITUTO FEDERAL
PARANÁ

LÓGICA DE PROGRAMAÇÃO

PROF^a. M.Sc. JULIANA H Q BENACCHIO

Modificadores de Tipos

- Os tipos de dados básicos em C podem estar acompanhados por modificadores na declaração de variáveis.
- Tais modificadores são: **long**, **short**, **signed** e **unsigned**.
- Os dois primeiros têm impacto no tamanho (número de bits) usados para representar um valor e os dois últimos indicam se o tipo será usado para representar valores negativos e positivos (**signed**) ou apenas positivos (**unsigned**).

Modificadores de Tipos

Tipo	Significado	Tamanho (bytes)	Intervalo
char	Caractere	1	-128 a 127
signed char	Caractere com sinal	1	-128 a 127
unsigned char	Caractere sem sinal	1	0 a 255
int	Inteiro	2	-32.768 a 32.767
signed int	Inteiro com sinal	2	-32.768 a 32.767
unsigned int	Inteiro sem sinal	2	0 a 65.535
short int	Inteiro curto	2	-32.768 a 32.767
signed short int	Inteiro curto com sinal	2	-32.768 a 32.767
unsigned short int	Inteiro curto sem sinal	2	0 a 65.535
long int	Inteiro longo	4	-2.147.483.648 a 2.147.483.647
signed long int	Inteiro longo com sinal	4	-2.147.483.648 a 2.147.483.647
unsigned long int	Inteiro longo sem sinal	4	0 a 4.294.967.295

Modificadores de Tipos

Tipo	Significado	Tamanho (bytes)	Intervalo
float	Ponto flutuante com precisão simples	4	$3.4 \cdot 10^{-38}$ a $3.4 \cdot 10^{38}$
double	Ponto flutuante com precisão dupla	8	$1.7 \cdot 10^{-308}$ a $1.7 \cdot 10^{308}$
long double	Ponto flutuante com precisão dupla longo	10	$3.4 \cdot 10^{-4932}$ a $1.1 \cdot 10^{4932}$

Inteiro (`int`)

- Um número inteiro é um número sem vírgula, que pode ser expresso em diferentes bases:
- **Base decimal:** o número inteiro é representado por uma sequência de números unitários (de 0 a 9), que não deve começar por 0.
- **Base hexadecimal:** o número inteiro é representado por uma sequência de números unitários (de 0 a 9 ou de A a F (ou de a a f)), começando por 0x ou 0X.
- **Base octal:** o número inteiro é representado por uma sequência de números unitários (incluindo apenas números de 0 a 7), começando com 0.

Número Inteiro (`int`)

- Os inteiros são assinados por padrão, o que significa que eles têm um sinal.
- Para armazenar informações sobre o sinal (em binário), os computadores usam o complemento de dois.

Ponto Flutuante (`float`/`double`)

- Um número com ponto flutuante é um número com vírgula, porém, ele pode ser representado de várias maneiras:
 - Um inteiro decimal: 895;
 - Um número com um ponto: 845.32;
 - Um número exponencial, ou seja, um número (eventualmente com vírgula) seguido da letra e (ou E) e de um inteiro correspondente à potência de 10 (assinado ou não, isto é, precedido de um + ou -):
 - 2.75e-2; 35.8E+10; .25e-2.

Ponto Flutuante (`float`/`double`)



- Na verdade, os números reais são números com ponto flutuante, ou seja, um número em que a posição da vírgula não é fixa e é identificada por uma parte de seus bits (conhecido como o expoente), o restante dos bits permitem codificar o número sem vírgula (a mantissa).

Ponto Flutuante (`float`/`double`)

- Os números do tipo `float` são codificados em 32 bits, incluindo:
 - 23 bits para a mantissa;
 - 8 bits para o expoente;
 - 1 bit para o sinal.

Ponto Flutuante (`float`/`double`)

- Os números do tipo **double** são codificados em 64 bits, incluindo:
 - 52 bits para a mantissa;
 - 11 bits para o expoente;
 - 1 bit para o sinal.

Ponto Flutuante (`float`/`double`)

- Os números do tipo `long double` são codificados em 80 bits, incluindo:
 - 64 bits para a mantissa;
 - 15 bits para o expoente;
 - 1 bit para o sinal.

Ponto Flutuante (`float`/`double`)



- A precisão dos números reais é aproximada. Ela depende do número de posições decimais, dependendo do tipo real, ela será, no mínimo:
 - de 6 números para o tipo **float**;
 - de 15 números para o tipo **double**;
 - de 17 números para o tipo **long double**.

Caractere (char)

- O tipo **char** (do inglês *character*) armazena o valor ASCII de um caractere, ou seja, um número inteiro.

ASCII control characters			ASCII printable characters			Extended ASCII characters										
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ù	161	í	193	ł	225	ô
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	õ
03	ETX	(End of Text)	35	#	67	C	99	c	131	á	163	ú	195	ł	227	ö
04	EOT	(End of Trans.)	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ø
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	†	229	ō
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	â	166	ª	198	ā	230	μ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Ă	231	þ
08	BS	(Backspace)	40	(72	H	104	h	136	ê	168	¿	200	Ē	232	þ
09	HT	(Horizontal Tab)	41)	73	I	105	i	137	ë	169	®	201	ƒ	233	ú
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	¬	202	ƒ	234	Û
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	í	171	½	203	ƒ	235	Ü
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¼	204	ƒ	236	ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ï	173	⅓	205	=	237	ÿ
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ā	174	«	206	ƒ	238	—
15	SI	(Shift In)	47	/	79	O	111	o	143	Ă	175	»	207	ƒ	239	'
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	⋮	208	ð	240	=
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	⋮	209	Đ	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	■	210	É	242	_
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ó	179		211	Ê	243	¼
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ô	180	└	212	Ë	244	¶
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	õ	181	┘	213	Ī	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	ù	182	┘	214	Ĳ	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ú	183	┘	215	ĳ	247	.
24	CAN	(Cancel)	56	8	88	X	120	x	152	ý	184	⊙	216	Ĵ	248	°
25	EM	(End of medium)	57	9	89	Y	121	y	153	ÿ	185	┘	217	Ķ	249	ˆ
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Û	186	┘	218	ĸ	250	˙
27	ESC	(Escape)	59	;	91	[123	{	155	ø	187	┘	219	■	251	˚
28	FS	(File separator)	60	<	92	\	124		156	ε	188	┘	220	■	252	˛
29	GS	(Group separator)	61	=	93]	125	}	157	ø	189	⊕	221	■	253	˜
30	RS	(Record separator)	62	>	94	^	126	~	158	x	190	∕	222	■	254	■
31	US	(Unit separator)	63	?	95	_			159	f	191	γ	223	■	255	nbsp

Caractere (**char**)

- Por padrão, os números são assinados, isso significa que eles têm um sinal. Para armazenar informações sobre o sinal (em binário), os computadores usam o complemento de dois.
- Então, um dado de tipo **char** é assinado, o que não significa que a letra possui um sinal, mas simplesmente que, na memória, o valor que codifica o caractere pode ser negativo.
- Se, por exemplo, quisermos armazenar a letra B (seu código ASCII é 66), poderemos definir este dado, seja pelo número 66, seja notando o 'B', onde as aspas simples significam **código ASCII de...**

Especificadores de formato

Tipo de Dados	<code>printf</code> / <code>scanf</code>
char	<code>%c</code>
int	<code>%d</code>
unsigned int	<code>%u</code>
short	<code>%hd</code>
unsigned short	<code>%hu</code>
long int	<code>%ld</code>
unsigned long int	<code>%lu</code>
float	<code>%f</code>
double	<code>%f</code> / <code>%lf</code>
long double	<code>%Lf</code>

Especificadores de formato

Saída	printf	Exemplo
Base Octal	<code>%o</code>	610
Base Hexadecimal	<code>%x</code>	7fa
Base Hexadecimal (maiúsculas)	<code>%X</code>	7FA
Notação Científica (exponencial)	<code>%e</code>	3.9265e+2
Notação Científica (exponencial maiúsculas)	<code>%E</code>	3.9265E+2
Geral, escolhe a mais curta entre <code>%f</code> e <code>%e</code>	<code>%g</code>	392.65
Hexadecimal ponto flutuante	<code>%a</code>	-0xc.90fep-2
Hexadecimal ponto flutuante (maiúsculas)	<code>%A</code>	-0XC.90FEP-2
String	<code>%s</code>	texto
Endereço (ponteiro)	<code>%p</code>	b8000000
Caractere %	<code>%%</code>	10%

Sequências de escape

Sequência de escape	Descrição
<code>\n</code>	Nova linha. Posiciona o cursor da tela no início da próxima linha
<code>\t</code>	Tabulação horizontal. Move o cursor da tela para a próxima posição de tabulação
<code>\a</code>	Alerta. Faz soar o alarme do sistema
<code>\\</code>	Barra invertida. Insere um caractere de barra invertida em uma string
<code>\"</code>	Aspas. Insere um caractere de aspas em uma string

Formatos de impressão (`printf`)



- Por exemplo, podemos especificar o tamanho do número utilizado para impressão da seguinte forma:
- `%6d` → inteiro, com pelo tamanho pelo menos 6
- `%6f` → ponto flutuante, com tamanho pelo menos 6
- `%.3f` → ponto flutuante, com 3 digitos depois do ponto decimal
- `%6.3f` → ponto flutuante, com tamanho pelo menos 6 e 3 digitosdepois do ponto decimal
- `%6.0f` → ponto flutuante, com pelo menos tamanho 6 e nenhum digitodepois do ponto decimal.