Programación de Computadores

Tema 2: Tipos de datos y representación a nivel de bits

Carrera Ingeniería Civil en Informática
y Ciencias de la Computación
Universidad de Concepción

José Fuentes - jfuentess@inf.udec.cl

Primer programa

Ver primer_programa.c

```c
#include <stdio.h>

int main() {
   int c1 = 4; // cateto 1
   int c2 = 3; // cateto 2
   int hp = 5; // hipotenusa

   // Verificamos si cumplen el teorema de Pitágoras
   if((c1*c1 + c2*c2) == hp*hp)
      printf("Se cumple el teorema de Pitágoras");

   return 0;
}
```

Equivalente en Python

```python
c1 = 4 # cateto 1
c2 = 3 # cateto 2
hp = 5 # hipotenusa

if((c1*c1 + c2*c2) == hp*hp)
  print("...")
```

Palabras reservadas

auto	do	goto	signed	unsigned
break	double	if	sizeof	void
case	else	int	static	volatile
char	enum	long	struct	while
const	extern	register	switch	continue
float	return	typedef	default	for
short	union			

Nombres de variables

Para asignar nombres a variables, se deben considerar las siguientes reglas:

- Sólo se pueden usar letras [a-zA-Z], dígitos [0-9] y guión bajo [_]
- El nombre no puede iniciar con un dígito
- No se pueden usar espacios en blanco
- No puede ser el nombre de una palabra reservada

Ejemplo:

Nombres válidos	**Nombres inválidos**
• Var1 • _var_type	• 1Var • for

Variables: Enteros y reales

Ver limites.c, signed_unsigned.c y tipos_de_datos.c

Nota: Puede variar entre sistemas operativos

	Tipo	Tamaño	Rango de representación	
			Con signo (signed)	Sin signo (unsigned)
entero	char	1 byte	$[-2^7 - 2^7-1]$	$[0 - 2^8-1]$
	short	2 bytes	$[-2^{15} - 2^{15}-1]$	$[0 - 2^{16}-1]$
	int	4 bytes	$[-2^{31} - 2^{31}-1]$	$[0 - 2^{32}-1]$
	long	8 bytes	$[-2^{63} - 2^{63}-1]$	$[0 - 2^{64}-1]$
real	float	4 bytes	[1,2e-38 - 3,4e38]	
	double	8 bytes	[2,3e-308 - 1,7e308]	

Intuición sobre representación de números reales: https://evanw.github.io/float-toy/

Operadores aritméticos

Operador	Operación	Ejemplo	Resultado
+	Suma	x = 4.5 + 3; v = 4.5 + 3;	x = 7 v = 7.5
-	Resta	x = 4.5 - 3; v = 4.5 - 3;	x = 1 v = 1.5

*	Multiplicación	x = 4.5 * 3;
		v = 4.5 * 3;
		v = 4 * 3;
			x = 12
			v = 13.5
			v = 12.0

/	División	x = 4 / 3;
		v = 4.0 / 3;
		v = (float) 4 / 3;
			x = 1
			v = 1.33
			v = 1.33

%	Módulo	x = 15 % 2; v = (15 % 2) / 2;	x = 1 v = 0.0

Operadores aritméticos

Operador	Ejemplo	Equivalencia	
+=	x += 5	x = x + 5	
-=	y -= 3	y = y - 3	
*=	x *= 5	x = x * 5	
/=	y /= 3	y = y / 3	
%=	x %= 5	x = x % 5	
++	x=5; y=x++ x=5; y=++x	y = x; x+=1 x+=1; y = x	x=6; y=5 x=6; y=6
--	x=3; y=x-- x=3; y=--x	y = x; x-=1 x-=1; y = x	x=2; y=3 x=2; y=2

Operadores lógicos y relacionales

	Operador	Operación	Ejemplo	Resultado
Lógicos	!	Negación	x = (!(7>15))	x = 1
	&&	Conjunción	y = (35 > 20) && 0	y = 0
	||	Disyunción	x = (35 > 20) || 0	x = 1
Relacionales	==	Igual a	r = (3 == 4)	r = 0
	!=	Diferente de	r = (3 != 4)	r = 1
	<	Menor que	r = (3 < 4)	r = 1
	>	Mayor que	r = (3 > 4)	r = 0
	<=	Menor o igual que	r = (3 <= 4)	r = 1
	>=	Mayor o igual que	r = (3 >= 4)	r = 0

Precedencia de operadores

Ver operadores.c

Operador	Función	Asociatividad
() [] ., ->, ++, --	LLamada función/Paréntesis Acceso a un arreglo Selección de elemento Post incremento/decremento	Izq -> Der
++, -- ! ~ *, &	Pre incremento/decremento Negación lógica Complemento a nivel de bit Dereferencia/referencia a puntero	Der -> Izq
*,/,%	Multiplicación, división, módulo	Izq -> Der
+,-	Suma, resta	Izq -> Der
<<,>>	Corrimiento de bits	Izq -> Der
<,<=,>=,>	Operadores relacionales	Izq -> Der

Precedencia

```
100 / 10 * 10

10 * 10
100
```

```
100 + 200 / 10 - 3 * 10

100 + 20 - 3 * 10
100 + 20 - 30
120 - 30
90
```

```
x=3; y=4
x++ * --y

3 * 3
9
```

```
10 + 8 * (20 - 3)

10 + 8 * 17
10 + 136
146
```

Constantes: const y #define

Ver celsius2Kelvin.c

```
const <tipo dato> <nombre> = <valor>;
```

```
const int nu = 20;      // constante de tipo entero
const float fl = 2.18; // constante de tipo flotante
const char ch = ‘f’;    // constante de tipo caracter
```

```
#define <nombre> <texto de reemplazo>
```

```
 #define nu 20             // Cada aparición de nu es reemplazada por 20
 #define fl 2.18           // Cada aparición de fl es reemplazada por 2.18
 #define ch ‘f’            // Cada aparición de ch es reemplazada por ‘f’
 #define formula (12 % 3) // Cada aparición de formula es reemplazada por (12%3)
```

Tabla ASCII (American Standard Code for Information Interchange)

— — —

ASCII control characters

DEC	HEX	Simbolo ASCII	
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de linea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift Out)
15	0Fh	SI	(shift In)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	20h	DEL	(delete)

ASCII printable characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`
33	21h	!	65	41h	A	97	61h	a
34	22h	"	66	42h	B	98	62h	b
35	23h	#	67	43h	C	99	63h	c
36	24h	$	68	44h	D	100	64h	d
37	25h	%	69	45h	E	101	65h	e
38	26h	&	70	46h	F	102	66h	f
39	27h	'	71	47h	G	103	67h	g
40	28h	(	72	48h	H	104	68h	h
41	29h	)	73	49h	I	105	69h	i
42	2Ah	*	74	4Ah	J	106	6Ah	j
43	2Bh	+	75	4Bh	K	107	6Bh	k
44	2Ch	,	76	4Ch	L	108	6Ch	l
45	2Dh	-	77	4Dh	M	109	6Dh	m
46	2Eh	.	78	4Eh	N	110	6Eh	n
47	2Fh	/	79	4Fh	O	111	6Fh	o
48	30h	0	80	50h	P	112	70h	p
49	31h	1	81	51h	Q	113	71h	q
50	32h	2	82	52h	R	114	72h	r
51	33h	3	83	53h	S	115	73h	s
52	34h	4	84	54h	T	116	74h	t
53	35h	5	85	55h	U	117	75h	u
54	36h	6	86	56h	V	118	76h	v
55	37h	7	87	57h	W	119	77h	w
56	38h	8	88	58h	X	120	78h	x
57	39h	9	89	59h	Y	121	79h	y
58	3Ah	:	90	5Ah	Z	122	7Ah	z
59	3Bh	;	91	5Bh	[	123	7Bh	{
60	3Ch	<	92	5Ch	\	124	7Ch	|
61	3Dh	=	93	5Dh	]	125	7Dh	}
62	3Eh	>	94	5Eh	^	126	7Eh	~
63	3Fh	?	95	5Fh	_			

theASCIIcode.com.ar

Extended ASCII characters

DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	└	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	┴	225	E1h	ß
130	82h	é	162	A2h	ó	194	C2h	┬	226	E2h	Ô
131	83h	â	163	A3h	ú	195	C3h	├	227	E3h	Ò
132	84h	ä	164	A4h	ñ	196	C4h	─	228	E4h	õ
133	85h	à	165	A5h	Ñ	197	C5h	┼	229	E5h	Õ
134	86h	å	166	A6h	ª	198	C6h	ã	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	Ã	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	╚	232	E8h	Þ
137	89h	ë	169	A9h	®	201	C9h	╔	233	E9h	Ú
138	8Ah	è	170	AAh	¬	202	CAh	╩	234	EAh	Û
139	8Bh	ï	171	ABh	½	203	CBh	╦	235	EBh	Ù
140	8Ch	î	172	ACh	¼	204	CCh	╠	236	ECh	ý
141	8Dh	ì	173	ADh	¡	205	CDh	═	237	EDh	Ý
142	8Eh	Ä	174	AEh	«	206	CEh	╬	238	EEh	¯
143	8Fh	Å	175	AFh	»	207	CFh	¤	239	EFh	´
144	90h	É	176	B0h	░	208	D0h	ð	240	F0h	
145	91h	æ	177	B1h	▒	209	D1h	Ð	241	F1h	±
146	92h	Æ	178	B2h	▓	210	D2h	Ê	242	F2h	‗
147	93h	ô	179	B3h	│	211	D3h	Ë	243	F3h	¾
148	94h	ò	180	B4h	┤	212	D4h	È	244	F4h	¶
149	95h	ò	181	B5h	Á	213	D5h	ı	245	F5h	§
150	96h	û	182	B6h	Â	214	D6h	Í	246	F6h	÷
151	97h	ù	183	B7h	À	215	D7h	Î	247	F7h	¸
152	98h	ÿ	184	B8h	©	216	D8h	Ï	248	F8h	°
153	99h	Ö	185	B9h	╣	217	D9h	┘	249	F9h	¨
154	9Ah	Ü	186	BAh	║	218	DAh	┌	250	FAh	·
155	9Bh	ø	187	BBh	╗	219	DBh	█	251	FBh	¹
156	9Ch	£	188	BCh	╝	220	DCh	▄	252	FCh	³
157	9Dh	Ø	189	BDh	¢	221	DDh	¦	253	FDh	²
158	9Eh	×	190	BEh	¥	222	DEh	Ì	254	FEh	■
159	9Fh	ƒ	191	BFh	┐	223	DFh	▀	255	FFh	

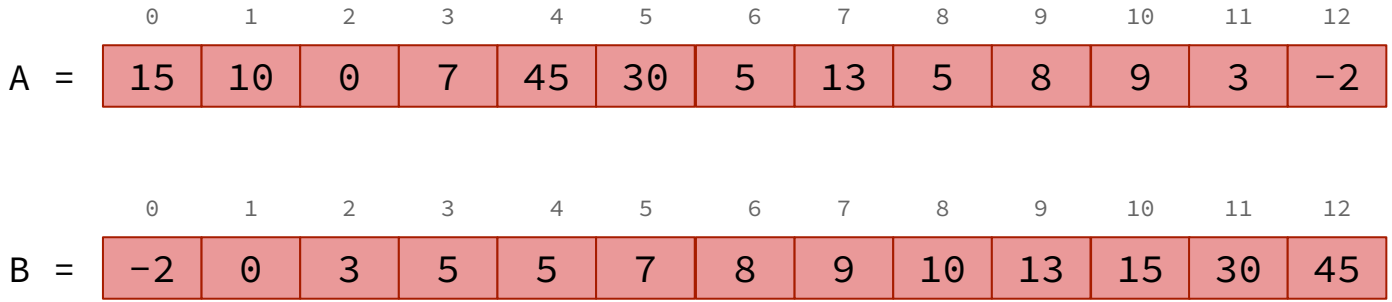
```
char x = '+';
int y  = x + 57;
/* y = 100 */
```

Ejercicio 1: Elemento central

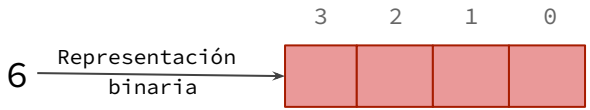
Ver elemento_central.c

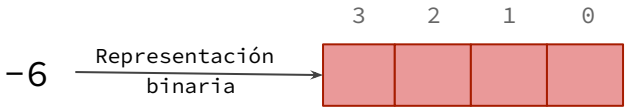
Ejercicio 2: Buscar en un arreglo

¿En cuál arreglo es más fácil buscar un valor?¿Por qué?



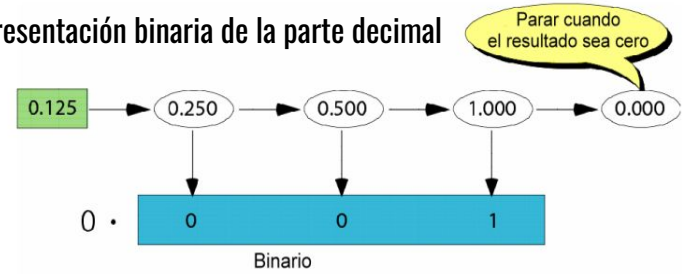
Representación binaria números enteros



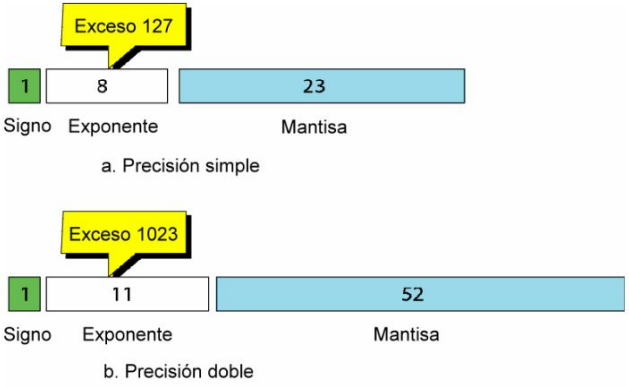


Representación de números punto flotante

Representación binaria de la parte decimal



Estándar IEEE (Institute of Electrical and Electronics Engineers)



Representación normalizada

Número original	Desplazamiento	Normalizado
+ 1010001.11001	← 6	$+2^{+6}$ x 1.01000111001
-111.000011	← 2	-2^{+2} x 1.11000011
+0.00000111001	6 →	$+2^{-6}$ x 1.11001
-0.001110011	3 →	-2^{-3} x 1.110011

Ejemplos estándar IEEE

Número	Signo	Exponente	Mantisa
-2^{2} x 1.11000011	1	10000001	11000011000000000000000
$+ 2^{-6}$ x 1.11001	0	01111001	11001000000000000000000
-2^{-3} x 1.110011	1	01111100	11001100000000000000000