

Appendix A

ANSI C Reserved Words

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

Special Character

Character	Name	Meaning
\b	Backspace	Move the cursor to the left one character.
\f	Form feed	Go to the top of a new page.
\n	Newline	Go to the next line.
\r	Return	Go to the beginning of the current line.
\t	Tab	Advance to the next tab stop (eight-column boundary).
\'	Apostrophe	The character '.
\"	Double quote	The character ".
\\	Backslash	The character \.
\nnn		The character number <i>nnn</i> (Octal)
\xhh		The character number <i>hh</i> (Hexadecimal)

Numeric Limits

Name	Bits	Low Value	High Value	Accuracy
int	16	-32768	32767	
short int	16	-32768	32767	
long int	32	-2147483648	2147483647	
unsigned int	16	0	65535	
unsigned short int	16	0	65535	
unsigned long int	32	0	4294967295	
char	8	-128	127	
unsigned char	8	0	255	
float	32	-3.4E+38	3.4E+38	6 digits
double	64	-1.7E+308	1.7E+308	15 digits
long double	80	-3.4E+4932	3.4E+4932	17 digits

ตารางที่ A.1 Precedence Rule

Precedence	Operator	Description	Associates from the
1	() [] -> . Dot Operator (struct member)	Function expression Array expression Arrow Operator (struct indirection) Dot Operator (struct member)	left ↓
2	++ -- ~ ! & * (type) - + sizeof	Increment, Decrement One's complement Negation (Not) Address Dereference Cast Unary minus Unary plus Size in bytes	right ↓
3	* / %	Multiplication Division Modulus	left
4	+ -	Addition Subtraction	
5	<< >>	Shift left Shift right	
6	< > <= >=	Less than Greater than Less than or equal Greater than or equal	
7	== !=	Equal Not equal	
8	&	Bitwise and	
9	^	Bitwise exclusive or	
10		Bitwise or	
11	&&	Logical and	
12		Logical or	↓
13	? :	Conditional	right
14	= %= += -= *= /= >>= <<= &= ^= =		↓
15	,	comma	left

Operator Precedence Rules

Precedence	Operators				
1	()	[]	->	.	(type)
2	! -(unary) &(address of)	~ *(dereference) sizeof	++	--	
3	*(multiply)	/	%		
4	+	-			
5	<<	>>			
6	<	<=	>	>=	
7	==	!=			
8	&(bitwise and)				
9	^				
10					
11	&&				
12					
13	?:				
14	=	+=	-=	etc.	
15	,				

Practical Subset

Precedence	Operators		
1	*(multiply)	/	%
2	+	-	