

Caratteri e stringhe in C

Codice ASCII a 7 bit

	0	16	32	48	64	80	96	112
	0x	1x	2x	3x	4x	5x	6x	7x
0 x0	NUL				@	P	`	p
1 x1					A	Q	a	q
2 x2					B	R	b	r
3 x3					C	S	c	s
4 x4					D	T	d	t
5 x5					E	U	e	u
6 x6					F	V	f	v
7 x7	BEL				G	W	g	w
8 x8	BS	(H	X	h	x
9 x9	TAB)			I	Y	i	y
10 xA	LF	*			J	Z	j	z
11 xB	ESC		+	;	K	[k	{
12 xC	FF	,			L	\	l	
13 xD	CR	-			M]	m	}
14 xE					N	^	n	~
15 xF					O	_	o	DEL

Variabili di tipo char

Definizione	<code>char ch ;</code>
Assegnazione	<code>ch = 'K' ;</code> <code>ch = 75 ;</code>
Lettura	<code>scanf("%c", &ch) ;</code> <code>ch = getchar() ;</code>
Stampa	<code>printf("%c", ch) ;</code> <code>putchar(ch) ;</code>

Sequenze di escape

'\n'	A capo
'\t'	Tabulazione
'\b'	Backspace (cancella ultimo carattere)
'\a'	Campanello (<i>alert</i>)
'\r'	Ritorno carrello sulla stessa riga
'\\'	Carattere di <i>backslash</i> \
'\''	Carattere di singolo apice '
'\"'	Carattere di doppio apice "
'\xNN'	Carattere il cui codice ASCII vale NN (in base 16)

Variabili di tipo stringa

Definizione	<code>char s[LUN+1] ;</code>
Assegnazione	<code>strcpy(s, "ciao") ;</code> <code>strcpy(s, s2) ;</code>
Lettura	<code>scanf("%s", s) ;</code> <code>gets(s) ;</code>
Stampa	<code>printf("%s", s) ;</code> <code>puts(s) ;</code>

Funzioni della libreria <string.h>

Nome	Parametri	Restituisce	Descrizione	Esempi
<code>strlen</code>	<code>char s[N]</code>	<code>int</code>	Lunghezza della stringa	<code>lun = strlen(s) ;</code>
<code>strcpy</code>	<code>char dst[N],</code> <code>char src[M]</code>		Copia il contenuto di <code>src</code> all'interno di <code>dst</code>	<code>strcpy(s1, s2) ;</code> <code>strcpy(s, "") ;</code> <code>strcpy(s1, "ciao") ;</code>
<code>strncpy</code>	<code>char dst[N],</code> <code>char src[M],</code> <code>int nc</code>		Copia il contenuto di <code>src</code> (max <code>nc</code> caratteri) all'interno di <code>dst</code>	<code>strncpy(s1, s2, 20) ;</code> <code>strncpy(s1, s2, MAX) ;</code>
<code>strcat</code>	<code>char dst[N],</code> <code>char src[N]</code>		Accoda il contenuto di <code>src</code> alla fine di <code>dst</code>	<code>strcat(s1, s2) ;</code> <code>strcat(s1, " ") ;</code>
<code>strncat</code>	<code>char dst[N],</code> <code>char src[M],</code> <code>int nc</code>		Accoda il contenuto di <code>src</code> (max <code>nc</code> caratteri) alla fine di <code>dst</code>	<code>strncat(s1, s2, 50) ;</code>
<code>strcmp</code>	<code>char s1[N],</code> <code>char s2[M]</code>	<code>int</code>	Risultato <0 se <code>s1</code> precede <code>s2</code> , =0 se <code>s1</code> è uguale a <code>s2</code> , >0 se <code>s1</code> segue <code>s2</code>	<code>if(strcmp(s, r)==0)</code> <code>while(strcmp(r,"")!=0)</code>
<code>strncmp</code>	<code>char s1[N],</code> <code>char s2[M],</code> <code>int n</code>	<code>int</code>	Come <code>strcmp</code> , ma confronta solo i primi <code>n</code> caratteri	<code>if(strncmp(r,</code> <code>"buon", 4)==0)</code>
<code>strchr</code>	<code>char s[N],</code> <code>char ch</code>	<code>==NULL</code> o <code>!=NULL</code>	Risultato <code>!=NULL</code> se il carattere <code>ch</code> compare nella stringa, <code>==NULL</code> se non compare.	<code>if(strchr(s, '.')!=NULL)</code> <code>if(strchr(s, ch)==NULL)</code>
<code>strspn</code>	<code>char s[N],</code> <code>char r[N]</code>	<code>int</code>	Restituisce la lunghezza della parte iniziale di <code>s</code> che è composta esclusivamente dei caratteri presenti in <code>r</code> (in qualsiasi ordine).	<code>lun = strspn(s, " ") ;</code> <code>lun = strspn(s, ",") ;</code>
<code>strcspn</code>	<code>char s[N],</code> <code>char r[N]</code>	<code>int</code>	Restituisce la lunghezza della parte iniziale di <code>s</code> che è composta esclusivamente dei caratteri non presenti in <code>r</code> .	<code>lun = strspn(s, " ") ;</code> <code>lun = strspn(s, ",") ;</code>

Funzioni della libreria <ctype.h>

Nome	Parametri	Restituisce	Descrizione	Esempi
isalpha	char ch	vero/falso	Lettera maiuscola o minuscola (A...Z, a... z)	<code>if(isalpha(ch))</code> { ... }
isupper	char ch	vero/falso	Lettera maiuscola (A... Z)	<code>if(isupper(ch))</code> { ... }
islower	char ch	vero/falso	Lettera minuscola (a... z)	<code>if(islower(ch))</code> { ... }
isdigit	char ch	vero/falso	Cifra numerica (0... 9)	<code>if(isdigit(ch))</code> { ... }
isalnum	char ch	vero/falso	Lettera oppure cifra numerica: <code>isalpha(ch) isdigit(ch)</code>	<code>if(isalnum(ch))</code> { ... }
isxdigit	char ch	vero/falso	Cifra numerica oppure lettera valida in base 16 (a...f, A... F)	<code>if(isxdigit(ch))</code> { ... }
ispunct	char ch	vero/falso	Simbolo di punteggiatura (!"#\$%&'()*+,-./:;<=>?@[\]^_`{ }~)	<code>if(ispunct(ch))</code> { ... }
isgraph	char ch	vero/falso	Qualsiasi simbolo visibile (lettera, cifra, punteggiatura)	<code>if(isgraph(ch))</code> { ... }
isprint	char ch	vero/falso	Qualsiasi simbolo visibile o spazio	<code>if(isprint(ch))</code> { ... }
isspace	char ch	vero/falso	Spazio, tab o a capo	<code>if(isspace(ch))</code> { ... }
isctrl	char ch	vero/falso	Qualsiasi carattere di controllo	<code>if(isctrl(ch))</code> { ... }
toupper	char ch	char	Ritorna la versione maiuscola di ch	<code>for(i=0; s[i]!=0; i++)</code> s[i] = toupper(s[i]);
tolower	char ch	char	Ritorna la versione minuscola di ch	<code>for(i=0; s[i]!=0; i++)</code> s[i] = tolower(s[i]);

