Χαρακτήρες

Χαρακτήρες - Εισαγωγή

- Έως τώρα έχουμε κατά κύριο λόγο χρησιμοποιήσει τους αριθμητικούς τύπους δεδομένων `int`, `float` και `double`
- Ο τύπος δεδομένων `char` είναι κι αυτός αριθμητικός
- Για τη διαχείριση των χαρακτήρων (και των αλφαριθμητικών στο επόμενο κεφάλαιο), θα θεωρήσουμε ότι το σύνολο των χαρακτήρων που υποστηρίζει ο υπολογιστής είναι κωδικοποιημένο σύμφωνα με το πιο διαδεδομένο πρότυπο, τον κώδικα ASCII, που αντιστοιχίζει κάθε χαρακτήρα σε μία αριθμητική τιμή
- Ο ASCII κώδικας αντιστοιχίζει (κωδικοποιεί) ένα σύνολο χαρακτήρων που αποτελείται από γράμματα, αριθμούς, σημεία στίξης, κτλ... με ακέραιες τιμές ανάμεσα στο `0` και το `255`
- Π.χ. η ASCII τιμή του χαρακτήρα `'C'` είναι το `67`, ενώ η ASCII τιμή του χαρακτήρα `'c'` είναι το `99`

Πίνακας ASCII (βασικοί χαρακτήρες)

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Πίνακας ASCII (επιπλέον χαρακτήρες)

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	φ
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	▉	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	^{n}
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	^{2}
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	

Ο τύπος `char` (I)

- Αφού ένας χαρακτήρας κωδικοποιείται σαν ακέραιος με τιμή ανάμεσα στο `0` και το `255`, ο τύπος δεδομένων `char`, που έχει μέγεθος 1 byte, μπορεί να χρησιμοποιηθεί για την αποθήκευση χαρακτήρων

- Στο επόμενο παράδειγμα δηλώνεται μία μεταβλητή τύπου `char` με το όνομα `ch` και αποθηκεύεται ο χαρακτήρας `'c'` σε αυτήν

```
char ch;
ch = 'c';
```

- Όταν χρησιμοποιείται κάποιος σταθερός χαρακτήρας πρέπει να περικλείεται σε μονές αποστρόφους (`' '`) και όχι σε διπλά εισαγωγικά

Ο τύπος `char` (II)

- Όταν αποθηκεύεται ένας χαρακτήρας σε μία μεταβλητή τύπου `char`, στην πραγματικότητα αποθηκεύεται η ASCII τιμή του χαρακτήρα
- Δηλαδή, στο προηγούμενο παράδειγμα στη μεταβλητή `ch` αποθηκεύτηκε η τιμή `99`
- Επομένως, οι εντολές:

```
ch = 'c';
```
και
```
ch = 99;
```
είναι ισοδύναμες

- Παρατηρήστε ότι οι πιο συνηθισμένοι χαρακτήρες, όπως γράμματα, ψηφία και σημεία στίξης αντιστοιχίζονται σε αριθμητικές τιμές ανάμεσα στο `0` και το `127`
- Οι χαρακτήρες με τιμές από `128` έως `255` αποτελούν το εκτεταμένο ASCII σύνολο και αντιστοιχίζονται σε εξεζητημένα γράμματα και ειδικά σύμβολα
- Επειδή η μέγιστη τιμή που μπορεί να πάρει μία μεταβλητή `char` είναι το `127` (θυμηθείτε ότι το εύρος τιμών του `char` είναι `-128...127`) σε περίπτωση που θέλουμε να αποθηκεύσουμε σε μία μεταβλητή `char` ένα χαρακτήρα με ASCII τιμή μεγαλύτερη από `127`, πρέπει να χρησιμοποιήσουμε τον τύπο `unsigned char` ή `int`

Εμφάνιση Χαρακτήρα

- Για την εμφάνιση ενός χαρακτήρα στην οθόνη (μέσω της `printf()`) χρησιμοποιείται το `%c`, ενώ για την εμφάνιση της ASCII τιμής του χρησιμοποιείται αντίστοιχα το `%d`
- Στο παράδειγμα δηλώνεται μία μεταβλητή τύπου `char` με το όνομα `ch` και αποθηκεύεται ο χαρακτήρας `'a'` σε αυτήν
- Στη συνέχεια εμφανίζεται στην οθόνη ο χαρακτήρας αυτός καθώς και η αντίστοιχη ASCII τιμή του, χρησιμοποιώντας τα προσδιοριστικά μετατροπής `%c` και `%d`, αντίστοιχα

```
#include <stdio.h>
int main()
{
	char ch;

	ch = 'a';
	printf("Char = %c and its ASCII code is %d\n",ch,ch);
	return 0;
}
```

- Ουσιαστικά, όταν ένας χαρακτήρας περιέχεται σε μία έκφραση – είτε είναι σταθερά είτε μεταβλητή – η C τον χειρίζεται σαν ακέραιο και χρησιμοποιεί την ASCII τιμή του

Παρατηρήσεις

- Ουσιαστικά, κάθε χαρακτήρας δεν είναι τίποτα άλλο παρά ένας μικρός ακέραιος αριθμός από 0 έως και 255
- Ανάλογα με το προσδιοριστικό μετατροπής που θα χρησιμοποιήσουμε (`%c` ή `%d`) εμφανίζεται ο ίδιος ο χαρακτήρας ή η ASCII τιμή του, αντίστοιχα

Παραδείγματα (Ι)

Dec	Hex	Cha
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z

- Ποια είναι η έξοδος του παρακάτω προγράμματος ???

```
#include <stdio.h>
int main()
{
        printf("Char = %c and its ASCII code = %d\n",'a'+3,'a'+3);
        return 0;
}
```

Έξοδος: `Char = d and its ASCII code = 100`

Παραδείγματα (ΙΙ)

- Γράψτε ένα πρόγραμμα το οποίο να εμφανίζει την ASCII τιμή του χαρακτήρα που αντιστοιχεί στην αλλαγή νέας γραμμής ή ισοδύναμα στο πάτημα του πλήκτρου Enter

```
#include <stdio.h>
int main()
{
	printf("ASCII code = %d\n",'\n');
	return 0;
}
```

Παραδείγματα (ΙΙΙ)

- Γράψτε ένα πρόγραμμα το οποίο να εμφανίζει στην οθόνη όλους τους χαρακτήρες και τις αντίστοιχες ASCII τιμές αυτών

```
#include <stdio.h>
int main()
{
	int i;
	for(i = 0; i < 256; i++)
		printf("Char = %c and its ASCII code = %d\n",i,i);
	return 0;
}
```

Παραδείγματα (IV)

- Παρατηρώντας προσεκτικά τον πίνακα με τις ASCII τιμές των χαρακτήρων, γράψτε ένα πρόγραμμα το οποίο να διαβάζει έναν κεφαλαίο χαρακτήρα και να εμφανίζει τον αντίστοιχο πεζό

```
#include <stdio.h>
int main()
{
	char ch;

	printf("Enter character: ");
	scanf("%c",&ch);

	printf("Char = %c\n",ch+32);
	return 0;
}
```

Η συνάρτηση `getchar()`

- Η συνάρτηση `getchar()` διαβάζει έναν χαρακτήρα από το `stdin` (το οποίο εξ' ορισμού συνδέεται με το πληκτρολόγιο)

- Το πρωτότυπό της δηλώνεται στο `stdio.h`, ως εξής:

```
int getchar();
```

- Αν η `getchar()` εκτελεστεί επιτυχημένα, επιστρέφει τον χαρακτήρα που διαβάστηκε

- Αν δεν υπάρχει άλλος χαρακτήρας στο `stdin` για να διαβαστεί ή αν συμβεί κάποιο λάθος, η `getchar()` επιστρέφει μία ειδική σταθερά που ονομάζεται `EOF` και έχει τιμή `-1`

- Η `getchar()` εκτελείται πιο γρήγορα από τη `scanf()`, γιατί η `scanf()` είναι μία σύνθετη συνάρτηση που έχει σχεδιαστεί για να διαβάζει διάφορους τύπους δεδομένων και όχι μόνο χαρακτήρες

Παράδειγμα

- Γράψτε ένα πρόγραμμα το οποίο να εμφανίζει και να μετράει τους χαρακτήρες που εισάγει ο χρήστης μέχρι να πατήσει το πλήκτρο Enter, με χρήση της συνάρτησης `getchar()`

```
#include <stdio.h>
int main()
{
	int ch,sum;

	printf("Enter characters: ");

	sum = 0;
	ch = getchar();
	while(ch != '\n')
	{
		sum++; /* Αυξάνεται η μεταβλητή που μετράει τους
χαρακτήρες που έχει εισάγει ο χρήστης μέχρι να διαβαστεί ο
χαρακτήρας '\n'. */
		printf("%c",ch);
		ch = getchar();
	}
	printf("\nTotal number is = %d\n",sum);
	return 0;
}
```