

Κεφάλαιο

5

Ο τύπος δεδομένων char



Ο τύπος δεδομένων char

Το κεφάλαιο αυτό εισάγει και αναλύει τον τρόπο χρήσης των χαρακτήρων (characters) στη γλώσσα C. Εξετάζουμε τον τρόπο με τον οποίο η C χειρίζεται τις σταθερές και τις μεταβλητές τύπου `char`. Ακομη, αναφερόμαστε και σε συναρτήσεις που χρησιμοποιούνται για το χειρισμό των χαρακτήρων.

Σταθερές τύπου char

Μια σταθερά τύπου `char` δεν είναι τίποτε άλλο από **έναν** χαρακτήρα που περικλείεται σε **μονά** εισαγωγικά. Για παράδειγμα, τα

`'ρ'`

`'1'`

`'*'`


είναι παραδείγματα σταθερών τύπου `char`.

Οι χαρακτήρες είναι αριθμοί!!!

Η εσωτερική απεικόνιση μιας σταθεράς (αλλά και μιας μεταβλητής) τύπου `char` είναι ένας αριθμός. Ο αριθμός αυτός είναι ο κωδικός ASCII⁵ του χαρακτήρα. Για παράδειγμα, η συνάρτηση

```
printf("%c %d\n", 'A', 'A');
```

θα εμφανίσει το A και το 65 (ο κωδικός ASCII του A).

 **Η C χειρίζεται τους χαρακτήρες σαν αριθμούς.** Αυτό σημαίνει ότι μπορούν να μετέχουν σε αριθμητικές παραστάσεις. Οι επόμενες παραστάσεις δεν είναι, λοιπόν, καθόλου παράξενες:

`5 + '*'` → επιστρέφει την τιμή 47 (5 + 42 που είναι ο κωδικός ASCII του '*').

`printf("%d %c\n", 'A'+1, 'A'+1);` → εμφανίζει στην οθόνη το 66 και το B.

⁵ Κάθε χαρακτήρας μετατρέπεται σε έναν αριθμό σύμφωνα με ένα καθορισμένο πρότυπο μετατροπής. Το πρότυπο αυτό ονομάζεται ASCII και ο ομώνυμος πίνακας μετατροπής υπάρχει στο τέλος του παρόντος κεφαλαίου.

Το 66 εμφανίζεται διότι είναι το αποτέλεσμα της παράστασης 'A'+1 και η παράμετρος %d "λέει" στην `printf()` να το εμφανίσει σαν ακέραιο αριθμό. Το B εμφανίζεται διότι η δεύτερη παράμετρος %c "λέει" στην `printf()` να εμφανίσει το αποτέλεσμα της παράστασης 'A'+1 (το 66) σαν χαρακτήρα. Έτσι εμφανίζεται το B, το οποίο έχει κωδικό ASCII 66.

Χαρακτήρες διαφυγής

Οι χαρακτήρες διαφυγής (escape characters) αναφέρονται και ως σειρές διαφυγής (escape sequences). Οι χαρακτήρες αυτοί δεν εμφανίζουν τίποτα στην οθόνη, αλλά ελέγχουν ορισμένες λειτουργίες της. Οι χαρακτήρες διαφυγής της C είναι:

<code>\n</code>	new line (νέα γραμμή)
<code>\b</code>	backspace (μία θέση πίσω)
<code>\f</code>	form feed (νέα σελίδα)
<code>\r</code>	carriage return (αρχή γραμμής)
<code>\t</code>	tab (επόμενη στηλοθετημένη στήλη)
<code>\\</code>	backslash (ο χαρακτήρας "ανάποδη κάθετος", \)
<code>\'</code>	τα απλά εισαγωγικά (')
<code>\"</code>	τα διπλά εισαγωγικά (")
<code>\0</code>	null character (byte 00000000)

Έτσι, η επόμενη συνάρτηση έχει το αποτέλεσμα στο διπλανό πλαίσιο:

Οι
"χαρακτήρες διαφυγής"
είναι 9

```
printf("Οι\n\"χαρακτήρες διαφυγής\"\\nείναι 9\n");
```

Παρατηρούμε ότι επειδή τα διπλά εισαγωγικά (") έχουν ειδικό νόημα για την `printf()`, δεν μπορούν να μουν στο αλφαριθμητικό μορφοποίησης σαν κανονικός χαρακτήρας. Για το λόγο αυτό πρέπει να χρησιμοποιηθεί ο χαρακτήρας διαφυγής \".

Μεταβλητές τύπου `char`

Μια μεταβλητή τύπου `char` δηλώνεται ως εξής:

`char` όνομα.μεταβλητής;

Οι μεταβλητές τύπου `char` δεσμεύουν 1 byte (8 bit). Στη μεταβλητή αποθηκεύεται ο κωδικός ASCII του χαρακτήρα, δηλαδή ένας ακέραιος αριθμός.

Η C χειρίζεται τις μεταβλητές χαρακτήρων σαν αριθμούς, οπότε μπορούν να μετέχουν σε αριθμητικές παραστάσεις.

Για παράδειγμα, μετά από τις παρακάτω προτάσεις:

```
int a,b;  
char ch;  
ch='A';  
a=ch+2;
```

η μεταβλητή `a` θα περιέχει στο τέλος τον αριθμό 67 (65+2).

Δύο συναρτήσεις χειρισμού χαρακτήρων

Για την είσοδο (από το πληκτρολόγιο) και την έξοδο (στην οθόνη) ενός χαρακτήρα χρησιμοποιούνται οι επόμενες συναρτήσεις:

`getch()`

Η συνάρτηση `getch()` περιμένει να πληκτρολογηθεί ένας χαρακτήρας και επιστρέφει ως τιμή τον κωδικό του χαρακτήρα. Δεν τον εμφανίζει στην οθόνη και δεν περιμένει πάτημα του πλήκτρου <Enter>.

Η συνάρτηση δεν χρειάζεται καμία παράμετρο και επιστρέφει ως τιμή έναν ακέραιο αριθμό, που είναι ο κωδικός ASCII του χαρακτήρα.

`putch(ch)`

Η συνάρτηση `putch()` εμφανίζει ένα χαρακτήρα στην οθόνη. Η παράμετρος `ch` πρέπει να είναι τύπου `char` ή τύπου `int` και ορίζει τον κωδικό ASCII του χαρακτήρα που θα εμφανιστεί. Ο χαρακτήρας εμφανίζεται στη θέση που βρίσκεται ο δρομέας (cursor). Στην περίπτωση που η παράμετρος είναι τύπου `int`

(4 byte) τότε ο χαρακτήρας που θα εμφανιστεί καθορίζεται από το πρώτο byte της παραμέτρου.

Συμβολοσειρές (character strings)

Μια συμβολοσειρά (ή "αλφαριθμητικό χαρακτήρων") ορίζεται στη C σαν ένα σύνολο χαρακτήρων μέσα σε διπλά εισαγωγικά `""`. Στη συνέχεια του βιβλίου οι συμβολοσειρές μπορεί να αναφέρονται και ως *σύνολα χαρακτήρων* αλλά και ως *αλφαριθμητικά* (character strings). Μερικά παραδείγματα συμβολοσειρών είναι:

`"ΝΙΚΟΣ"`

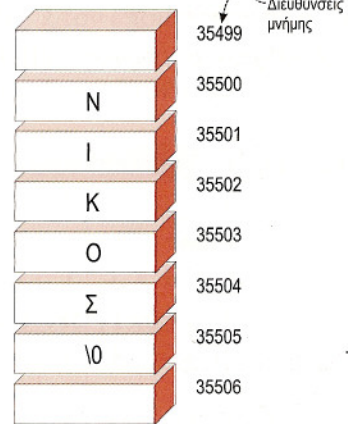
`"12"`

`""`


`"C is the Best."`

Στη μνήμη, μία συμβολοσειρά αποθηκεύεται σε συνεχόμενες θέσεις (μία για κάθε χαρακτήρα) ξεκινώντας από μια συγκεκριμένη διεύθυνση. Ο μεταγλωττιστής της C τοποθετεί έναν τερματικό χαρακτήρα `'\0'` στο τέλος κάθε συνόλου χαρακτήρων. Ο χαρακτήρας αυτός σηματοδοτεί το τέλος της συμβολοσειράς.

Εσωτερική απεικόνιση της συμβολοσειράς "ΝΙΚΟΣ"



Η τιμή του `"ΝΙΚΟΣ"` είναι η διεύθυνση της πρώτης θέσης μνήμης που έχουν αποθηκευτεί οι χαρακτήρες. Στη συγκεκριμένη περίπτωση η τιμή του `"ΝΙΚΟΣ"` είναι 35500.

 Όταν χρησιμοποιούνται σε ένα πρόγραμμα δύο ίδιες συμβολοσειρές, αποθηκεύονται δύο φορές σε διαφορετικές θέσεις της μνήμης.

Οι συμβολοσειρές έχουν τιμή;

Όπως κάθε παράσταση στη C, έτσι και οι συμβολοσειρές έχουν μία τιμή.

Η τιμή μιας **συμβολοσειράς** (string) είναι η **αρχική διεύθυνση** της μνήμης όπου αποθηκεύτηκε η συμβολοσειρά **και όχι** οι ίδιοι οι χαρακτήρες. Η C χειρίζεται κάθε συμβολοσειρά σαν αριθμό!!! και, πιο συγκεκριμένα, σαν μία διεύθυνση.

Η C διαθέτει στον προγραμματιστή μια σειρά από συναρτήσεις για το χειρισμό των συμβολοσειρών. Στις συναρτήσεις αυτές διαβιβάζεται πάντα σαν παράμετρος η αρχική διεύθυνση της συμβολοσειράς. Η συνάρτηση εντοπίζει το τέλος της συμβολοσειράς ανιχνεύοντας το χαρακτήρα τερματισμού '\0'.

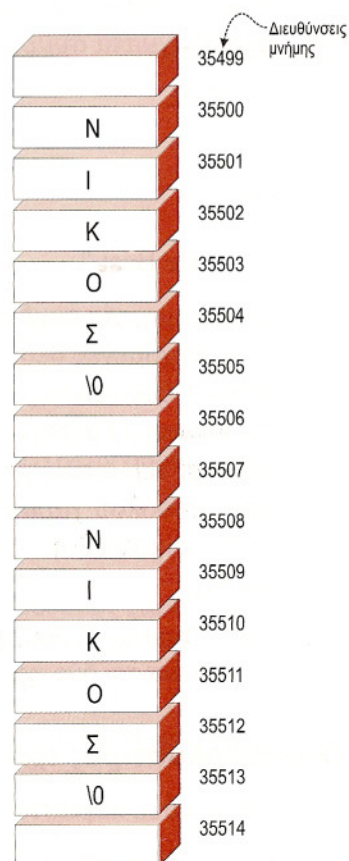
Αν προσπαθήσουμε να χειριστούμε ένα αλφαριθμητικό απευθείας, θα βρεθούμε μπροστά σε δυσάρεστες (αλλά όχι τόσο ακατανόητες) εκπλήξεις. Ας θεωρήσουμε το επόμενο πρόγραμμα, το οποίο σχεδιάστηκε για να συγκρίνει δύο ίδιες συμβολοσειρές.

```
main()
{
    if ("ΝΙΚΟΣ"=="ΝΙΚΟΣ")
    {
        printf("είναι ίδια\n");
    }
    else
    {
        printf("δεν είναι ίδια\n");
    }
}
```

και το αποτέλεσμα παραδόξως είναι:
δεν είναι ίδια

Αν τώρα αναλογιστούμε τη φιλοσοφία με την οποία η C χειρίζεται τις συμβολοσειρές θα διαπιστώσουμε ότι το παραπάνω αποτέλεσμα είναι απόλυτα λογικό.

Στη σύγκριση "ΝΙΚΟΣ"=="ΝΙΚΟΣ" η C δεν συγκρίνει τους χαρακτήρες των συμβολοσειρών αλλά τις τιμές τους. Προηγουμένως αναφέραμε, όμως, ότι ακόμη και οι ίδιες συμβολοσειρές αποθηκεύονται στη μνήμη τόσες φορές, όσες παρουσιάζονται μέσα στο πρόγραμμα —και μάλιστα σε διαφορετικές διευθύνσεις. Το παραπάνω σχήμα δείχνει μια υποθετική εικόνα της μνήμης για το συγκεκριμένο παράδειγμα. Η τιμή λοιπόν της μιας συμβολοσειράς είναι 35500 ενώ της άλλης 35508. Είναι φυσικό λοι-



πόν η σύγκριση "ΝΙΚΟΣ"=="ΝΙΚΟΣ" (ουσιαστικά 35500==35508) να είναι ψευδής.

✎ Για τη σύγκριση των συμβολοσειρών χρησιμοποιείται η συνάρτηση `strcmp()`, την οποία θα παρουσιάσουμε σε επόμενο κεφάλαιο, μαζί με άλλες συναρτήσεις που χειρίζονται συμβολοσειρές.

Πίνακας ASCII

Κάθε χαρακτήρας κωδικοποιείται σε έναν αντίστοιχο αριθμό. Η κωδικοποίηση αυτή ακολουθεί ορισμένα διεθνή πρότυπα (ASCII). Στον επόμενο πίνακα εμφανίζεται το επεκταμένο σύνολο χαρακτήρων ASCII με τους αντίστοιχους κωδικούς τους.

ASCII	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
32		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
96	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
128	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O	Π
144	P	Σ	T	Υ	Φ	X	Ψ	Ω	α	β	γ	δ	ε	ζ	η	θ
160	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	ς	τ	υ	φ	χ	ψ
176	▒	▒	▒													
192	Ł	ł	Ť	ť	—	†	‡	§	¶	§	§	§	§	=	≠	≠
208	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘
224	ω	ά	έ	ή	ϊ	ί	ό	ύ	ϋ	ώ	Α	Ε	Η	Ι	Ο	Υ
240	Ω	±	≥	≤	İ	ÿ	÷	≈	°	·	·	√	n	²	■	

Ο κωδικός ενός χαρακτήρα προκύπτει από το άθροισμα του αριθμού της γραμμής και της στήλης που βρίσκεται. Έτσι, ο χαρακτήρας "Γ" έχει κωδικό 130 (128+2).

Η τυπική κωδικοποίηση ASCII αφορά στους 128 πρώτους χαρακτήρες με κωδικούς από το 0 έως το 127. Οι χαρακτήρες από το 128 μέχρι το 255 αποτελούν

επέκταση του πρότυπου πίνακα χαρακτήρων και διαφέρουν από σύστημα σε σύστημα.

Οι χαρακτήρες με κωδικούς από 0 μέχρι 31 δεν είναι εκτυπώσιμοι χαρακτήρες και χρησιμοποιούνται ως χαρακτήρες ελέγχου. Χαρακτηριστικό παράδειγμα χαρακτήρων ελέγχου αποτελούν οι χαρακτήρες διαφυγής της C.

Παραδείγματα

Π.1 Το επόμενο πρόγραμμα διαβάζει και εμφανίζει ένα χαρακτήρα.

```
main()
{
    char c;
    printf("Πληκτρολογήστε ένα χαρακτήρα:");
    c = getch();
    putchar(c);
    putchar('\n');
}
```

Π.2 Το επόμενο πρόγραμμα διαβάζει ένα χαρακτήρα και τυπώνει τον ASCII κωδικό του.

```
main()
{
    char c;
    printf("Πληκτρολογήστε ένα χαρακτήρα:");
    c = getch();
    printf("ο κωδικός ASCII του %c είναι %d\n", c, c);
}
```

Π.3 Το επόμενο πρόγραμμα περιμένει το πάτημα ενός χαρακτήρα και ελέγχει αν είναι ελληνικός κεφαλαίος ή άλλος.

```
main()
{
```



```

char c;
printf("Πληκτρολογήστε ένα χαρακτήρα:");
c = getch();
if ('A'<=c && c<='Ω')
{
    printf("ο χαρακτήρας είναι ελληνικός κεφαλαίος\n");
}
else
{
    printf("ο χαρακτήρας δεν είναι ελληνικός κεφαλαίος\n");
}
}

```

Π.4 Παρατηρήστε το αποτέλεσμα του επόμενου προγράμματος.

```

main()
{
    char ch,let;
    int a,b=4;
    ch='A';
    let='B';
    a=b+ch+let;
    ch++;
    printf("a=%d ch=%c\n",a,ch);
}

```

a=135 ch=B

Η μεταβλητή **a** θα πάρει την τιμή 135 ($4+65+66$), δεδομένου ότι ο κωδικός του 'A' (στη **ch**) είναι 65 και του 'B' (στη **let**) είναι 66.

Με την παράσταση **ch++** η τιμή του **ch** θα γίνει 66, οπότε στην **printf()** θα εμφανιστεί το 'B'.

Ανασκόπηση Κεφαλαίου 5

- Ο τύπος δεδομένων `char` αφορά χαρακτήρες. Οι μεταβλητές τύπου `char` δεσμεύουν 1 byte.
- Η εσωτερική απεικόνιση μιας σταθεράς (αλλά και μιας μεταβλητής) τύπου `char` είναι ένας αριθμός. Ο αριθμός αυτός είναι ο κωδικός ASCII του χαρακτήρα.
- Η C χειρίζεται τους χαρακτήρες σαν ακέραιους αριθμούς. Μπορούν να μετέχουν σε πράξεις και αριθμητικές παραστάσεις.
- Εκτενώς χρησιμοποιούνται και ειδικοί χαρακτήρες, όπως ο χαρακτήρας αλλαγής γραμμής `\n` και ο χαρακτήρας τερματισμού `\0`.
- Η συνάρτηση `getch()` περιμένει να πληκτρολογηθεί ένας χαρακτήρας και επιστρέφει ως τιμή αυτόν το χαρακτήρα.
- Η συνάρτηση `putch(ch)` εμφανίζει το χαρακτήρα `ch` στην οθόνη.
- Οι συμβολοσειρές (ή *αλφαριθμητικά*) αποτελούνται από χαρακτήρες αποθηκευμένους σε συνεχόμενες θέσεις μνήμης, οι οποίες τερματίζονται με τον ειδικό χαρακτήρα `\0`.
- Η C "βλέπει" τις συμβολοσειρές σαν αριθμούς! Η τιμή μιας συμβολοσειράς είναι η αρχική διεύθυνση της μνήμης όπου αποθηκεύτηκε το σύνολο χαρακτήρων και όχι οι ίδιοι οι χαρακτήρες.
- Για το χειρισμό των συμβολοσειρών χρησιμοποιούνται ειδικές συναρτήσεις.

Ασκήσεις Κεφαλαίου 5

- 5.1 Να γραφεί πρόγραμμα το οποίο να δέχεται ένα χαρακτήρα από το πληκτρολόγιο και να τον επεξεργάζεται ως εξής: ★★
- Αν ο χαρακτήρας είναι πεζός, να τον τυπώνει στην οθόνη.
 - Εάν πρόκειται για αριθμητικό ψηφίο (0-9), να εμφανίζει το μήνυμα: "Πατήθηκε ένα ψηφίο".
 - Σε κάθε άλλη περίπτωση να μην κάνει τίποτα.

5.2 Να γραφεί πρόγραμμα το οποίο να δέχεται χαρακτήρες από το πληκτρολόγιο και να τους επεξεργάζεται ως εξής: ★★

- Αν ο χαρακτήρας είναι αριθμητικός (ψηφίο) να τον εμφανίζει όπως είναι.
- Αν είναι αλφαβητικός (είτε πεζός είτε κεφαλαίος) να εμφανίζει τον αμέσως επόμενο χαρακτήρα (πχ αν πληκτρολογηθεί α να εμφανιστεί το β κ.ο.κ.).

5.3 Να γραφεί πρόγραμμα το οποίο να εμφανίζει το επόμενο "μενού", να δέχεται έναν αριθμό, και να κάνει την αντίστοιχη ενέργεια. ★★★

1-Εκτύπωσε την λέξη "Hello"

2-Εκτύπωσε τον αριθμό 2

3-Εκτύπωσε "bye bye"

4-Μην κάνεις τίποτα

Δώσε επιλογή:

Αν η επιλογή δεν είναι αποδεκτή (εκτός από 1, 2, 3, 4), να εμφανίζεται το μήνυμα "Λάθος επιλογή".

5.4 Τι αποτέλεσμα θα έχει το παρακάτω πρόγραμμα; ★★

```
main()
{
    char ch,b='A';
    ch='A';
    if(ch==b)
        printf("NAI-1");
    else
        printf("OXI-1");
    if("A"=="A")
        printf("NAI-2");
    else
        printf("OXI-2");
}
```


5.5 Ποια από τα παρακάτω αληθεύουν: ★

- ☐ Μπορούμε να χειριζόμαστε τους χαρακτήρες σαν αριθμούς.
- ☐ Μια μεταβλητή τύπου `char` έχει μέγεθος ενός byte.
- ☐ Σε μια μεταβλητή χαρακτήρα δεν μπορούμε να καταχωρίσουμε έναν αριθμό.
- ☐ Οι συμβολοσειρές προσδιορίζονται από τη διεύθυνση όπου είναι αποθηκευμένος ο πρώτος τους χαρακτήρας.
- ☐ Μια συμβολοσειρά καταλαμβάνει τόσα byte όσοι **ακριβώς** είναι και οι χαρακτήρες που περιέχει.

5.6 Να γραφεί πρόγραμμα το οποίο να εμφανίζει τους κωδικούς των χαρακτήρων 'a', '*' και του κενού διαστήματος ' '. ★★

5.7 Να γραφεί πρόγραμμα το οποίο να εμφανίζει τους χαρακτήρες με κωδικούς ASCII 80, 125, και 192. ★★

5.8 Ποιο θα είναι το αποτέλεσμα του επόμενου προγράμματος: ★★

```
main()
{
    char ch=68, let='L';
    int a=2, b=4;
    a=ch+let;
    ch=++let;
    printf("a=%d ch=%c let=%c\n", ++a, ch, let);
}
```

5.9 Ποια η διαφορά του 'A' με το "A". ★★