

Η ΓΛΩΣΣΑ C

Μάθημα 10:

Χαρακτήρες και Συμβολοσειρές

Δημήτρης Ψούνης



www.psounis.gr



Περιεχόμενα Μαθήματος

A. Χαρακτήρες

1. Ο τύπος δεδομένων char
2. Ο πίνακας ASCII
3. Χρήση Μεταβλητών τύπου char

B. Συμβολοσειρές

1. Τι είναι η συμβολοσειρά
2. Διάβασμα και Εκτύπωση Συμβολοσειράς
3. Οι συναρτήσεις gets και puts
4. Η βιβλιοθήκη string.h

Γ. Ασκήσεις

A. Χαρακτήρες

1. Ο τύπος δεδομένων char

- Στην C ο τύπος δεδομένων char χρησιμοποιείται για να αποθηκεύσουμε χαρακτήρες.
- Στην πραγματικότητα όμως δεν αποθηκεύονται χαρακτήρες! Όλα τα δεδομένα αποθηκεύονται σε αριθμητική μορφή!
- Έτσι και οι χαρακτήρες είναι ακέραιοι αριθμοί από το 0 έως το 255.
 - Και χρησιμοποιείται ένας μεταφραστικός πίνακας, ο λεγόμενος πίνακας των ASCII κωδικών, που κάνει την αντιστοίχιση σε κάθε αριθμό με τον αντίστοιχο χαρακτήρα
 - Όποτε ο μεταγλωτιστής θέλει να διαχειριστεί έναν χαρακτήρα, συσχετίζει τον χαρακτήρα με τον αριθμό που αντιστοιχεί σε αυτόν, σύμφωνα με τον πίνακα ASCII.
- Άρα θα πρέπει να ξέρουμε ότι:
 - Αν μια μεταβλητή char χρησιμοποιηθεί ως χαρακτήρας, ερμηνεύεται ως χαρακτήρας
 - Αν μια μεταβλητή char χρησιμοποιηθεί ως αριθμός, ερμηνεύεται ως αριθμός



A. Χαρακτήρες

2. Ο πίνακας ASCII

- Στο συνημμένο αρχείο μπορείτε να βρείτε τον πίνακα χαρακτήρων ASCII. Το ενδιαφέρον μας θα εστιαστεί στη στήλη που έχει τους χαρακτήρες και τον αντίστοιχο αύξοντα αριθμό του χαρακτήρα.

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
^@	0	00		NUL	32	20		64	40	@	96	60	'	128	80	Ç	160	A0	À	192	C0	Ā	224	E0	α
^A	1	01		SOH	33	21	!	65	41	Α	97	61	a	129	81	Ć	161	A1	Á	193	C1	Ā	225	E1	β
^B	2	02		STX	34	22	"	66	42	Β	98	62	b	130	82	Ĉ	162	A2	Â	194	C2	Ĳ	226	E2	γ
^C	3	03		ETX	35	23	#	67	43	Γ	99	63	c	131	83	Č	163	A3	Č	195	C3	Ĵ	227	E3	δ
^D	4	04		EOT	36	24	\$	68	44	Δ	100	64	d	132	84	Ċ	164	A4	Ď	196	C4	Ĺ	228	E4	ε
^E	5	05		ENQ	37	25	%	69	45	Ε	101	65	e	133	85	Ď	165	A5	Ě	197	C5	Ļ	229	E5	ς
^F	6	06		ACK	38	26	&	70	46	Ε	102	66	f	134	86	Ě	166	A6	Ě	198	C6	ŀ	230	E6	ζ
^G	7	07		BEL	39	27	'	71	47	Ζ	103	67	g	135	87	Ħ	167	A7	Ő	199	C7	ŀ	231	E7	η
^H	8	08		BS	40	28	(72	48	Η	104	68	h	136	88	Ħ	168	A8	Ű	200	C8	ŀ	232	E8	θ
^I	9	09		HT	41	29)	73	49	Θ	105	69	i	137	89	İ	169	A9	Ű	201	C9	ŀ	233	E9	ι
^J	10	0A		LF	42	2A	*	74	4A	Ι	106	6A	j	138	8A	İ	170	AA	Ű	202	CA	ŀ	234	EA	κ
^K	11	0B		VT	43	2B	+	75	4B	Κ	107	6B	k	139	8B	İ	171	AB	Ű	203	CB	ŀ	235	EB	λ
^L	12	0C		FF	44	2C	,	76	4C	Λ	108	6C	l	140	8C	İ	172	AC	Ű	204	CC	ŀ	236	EC	μ
^M	13	0D		CR	45	2D	-	77	4D	Μ	109	6D	m	141	8D	İ	173	AD	Ű	205	CD	ŀ	237	ED	ν
^N	14	0E		SO	46	2E	.	78	4E	Ν	110	6E	n	142	8E	İ	174	AE	Ű	206	CE	ŀ	238	EE	ξ
^O	15	0F		SI	47	2F	/	79	4F	Ξ	111	6F	o	143	8F	İ	175	AF	Ű	207	CF	ŀ	239	EF	ο
^P	16	10		DLE	48	30	0	80	50	Ο	112	70	o	144	90	İ	176	BO	Ű	208	DO	ŀ	240	FO	π
^Q	17	11		DC1	49	31	1	81	51	Ο	113	71	q	145	91	İ	177	B1	Ű	209	D1	ŀ	241	F1	ρ
^R	18	12		DC2	50	32	2	82	52	Ρ	114	72	r	146	92	İ	178	B2	Ű	210	D2	ŀ	242	F2	σ
^S	19	13		DC3	51	33	3	83	53	Σ	115	73	s	147	93	İ	179	B3	Ű	211	D3	ŀ	243	F3	τ
^T	20	14		DC4	52	34	4	84	54	Τ	116	74	t	148	94	İ	180	B4	Ű	212	D4	ŀ	244	F4	υ
^U	21	15		NAK	53	35	5	85	55	Υ	117	75	u	149	95	İ	181	B5	Ű	213	D5	ŀ	245	F5	φ
^V	22	16		SYN	54	36	6	86	56	Υ	118	76	v	150	96	İ	182	B6	Ű	214	D6	ŀ	246	F6	χ
^W	23	17		ETB	55	37	7	87	57	Φ	119	77	w	151	97	İ	183	B7	Ű	215	D7	ŀ	247	F7	ψ
^X	24	18		CAN	56	38	8	88	58	Ψ	120	78	x	152	98	İ	184	B8	Ű	216	D8	ŀ	248	F8	ω
^Y	25	19		EM	57	39	9	89	59	Ω	121	79	y	153	99	İ	185	B9	Ű	217	D9	ŀ	249	F9	κ
^Z	26	1A		SUB	58	3A	:	90	5A	Ζ	122	7A	z	154	9A	İ	186	BA	Ű	218	DA	ŀ	250	FA	λ
^[27	1B		ESC	59	3B	;	91	5B	Ζ	123	7B	{	155	9B	İ	187	BB	Ű	219	DB	ŀ	251	FB	μ
^\	28	1C		FS	60	3C	<	92	5C	Ζ	124	7C	[156	9C	İ	188	BC	Ű	220	DC	ŀ	252	FC	ν
^]	29	1D		GS	61	3D	=	93	5D	Ζ	125	7D]	157	9D	İ	189	BD	Ű	221	DD	ŀ	253	FD	ξ
^^	30	1E		RS	62	3E	>	94	5E	Ζ	126	7E	^	158	9E	İ	190	BE	Ű	222	DE	ŀ	254	FD	ο
^_	31	1F		US	63	3F	?	95	5F	Ζ	127	7F	_	159	9F	İ	191	BF	Ű	223	DF	ŀ	255	FF	ς

A. Χαρακτήρες

3. Χρήση Μεταβλητών τύπου char

- Δηλώνουμε έναν χαρακτήρα με την συνήθη εντολή δήλωσης:

```
char ch;
```

- Για να αναθέσουμε στο χαρακτήρα ch την τιμή π.χ. του χαρακτήρα a μπορούμε να το κάνουμε με δύο τρόπους:

- Είτε μέσω του αντίστοιχου ASCII κωδικού:

```
ch=97;
```

- Είτε μέσω συμβολικής απεικόνισης:

```
ch='a';
```

- Προσοχή! Όταν θέλουμε να απεικονίσουμε έναν χαρακτήρα, θα πρέπει υποχρεωτικά να τον γράψουμε μέσα σε μονά εισαγωγικά!

- Επίσης ο προσδιοριστής μετατροπής της printf για τον τύπο δεδομένων χαρακτήρα είναι %c. Έτσι θα μπορούσαμε να τυπώσουμε τον χαρακτήρα με την εντολή:

```
printf("%c", ch);
```

A. Χαρακτήρες

3. Χρήση Μεταβλητών τύπου char

- Τρέξτε το ακόλουθο πρόγραμμα που δείχνει πως απεικονίζεται οι χαρακτήρες σε ένα πρόγραμμα C

```
/* char.c */
#include <stdio.h>

main()
{
    char c;

    for (c=80; c<=100; c++)
    {
        printf("\nO xaraktiras %d einai %c", c, c);
    }
}
```

- Σημειώστε ότι για να εκτυπωθούν οι ASCII 128-255 πρέπει να δηλώσουμε την μεταβλητή c σαν unsigned char.

B. Συμβολοσειρές

1. Τι είναι η συμβολοσειρά

- Σε πολλές περιπτώσεις για να προβάλλουμε κείμενο θέλουμε να απεικονίσουμε πολλούς χαρακτήρες σε σειρά. Αυτή είναι η έννοια της συμβολοσειράς, δηλαδή είναι μια σειρά από χαρακτήρες.

- Στην C αυτό γίνεται δηλώνοντας απλά έναν πίνακα από χαρακτήρες. Έτσι η εντολή δήλωσης:

```
char str[20];
```

- Δηλώνει έναν πίνακα από 20 χαρακτήρες, άρα μια συμβολοσειρά το πολύ 19 χαρακτήρων.

- ΠΡΟΣΟΧΗ! Είναι το πολύ 19 χαρακτήρων (και όχι 20) διότι πάντα σε μια συμβολοσειρά απεικονίζεται το τέλος της συμβολοσειράς με τον ειδικό χαρακτήρα \0 (slash μηδέν).

- Για παράδειγμα η συμβολοσειρά «hello» αποθηκεύεται στην μνήμη ως εξής:

str[0]	str[1]	str[2]	str[3]	str[4]	str[5]	str[20]
h	e	l	l	o	\0	...

B. Συμβολοσειρές

1. Τι είναι η συμβολοσειρά

- Η αρχικοποίηση μιας συμβολοσειράς μπορεί να γίνει κατά την δήλωση με τρεις τρόπους:

- Με τον γνωστό τρόπο αρχικοποίησης διάταξης:

```
char pin[3]={'a', 'b', '\0'};
```

- Επίσης με έναν συντομογραφικό τρόπο, χρησιμοποιώντας τα διπλά εισαγωγικά

```
char pin[3]="ab";
```

- Και με έναν ακόμη τρόπο που δεσμεύει τον απαραίτητο χώρο (στο παράδειγμα 3 θέσεις μνήμης:

```
char *pin="ab";                    ή    char pin[]="ab";
```

- Προσοχή όμως ότι αν δηλώσουμε και αρχικοποιήσουμε την συμβολοσειρά μέσω δείκτη, τότε δεν μπορούμε να τροποποιήσουμε το περιεχόμενό της (συμπεριφέρεται σαν σταθερά)

B. Συμβολοσειρές

2. Διάβασμα και Εκτύπωση μίας Συμβολοσειράς

- Για την εκτύπωση (με την printf) και την ανάγνωση (με την scanf) μιας συμβολοσειράς χρησιμοποιείται ο προσδιοριστής %s.
- Ωστόσο πρέπει να είμαστε προσεκτικοί!
 - Μία συμβολοσειρά είναι ένας πίνακας χαρακτήρων, άρα αφού είναι πίνακας, το όνομα της συμβολοσειράς είναι δείκτης στην αρχή της διάταξης.
 - Έτσι στην εντολή scanf δεν πρέπει να βάλουμε το & στο όνομα της μεταβλητής.
- Για παράδειγμα αν έχουμε δηλώσει μία συμβολοσειρά:

```
char string[100];
```

- Τότε η εκτύπωσή της θα γίνεται με την εντολή:

```
printf("%s", string);
```

- Ενώ το διάβασμα της συμβολοσειράς θα γίνεται με την εντολή:

```
scanf("%s", string);
```

B. Συμβολοσειρές

2. Διάβασμα και Εκτύπωση μίας Συμβολοσειράς

- Τρέξτε το ακόλουθο πρόγραμμα και εισάγετε στην είσοδο πρώτα την συμβολοσειρά «teststring» και έπειτα την συμβολοσειρά «test string» (πρώτα χωρίς κενό και μετά με κενό) και δείτε το αποτέλεσμα.

```
/* string.c: Deixnei tin xrisi tis scanf kai tis printf me
simvoloseires */
#include <stdio.h>

main()
{
    char string[80];

    printf("Dwste mia simboloseira: ");
    scanf("%s", string);

    printf("Pliktrologisate tin simvoloseira: %s", string);
}
```

B. Συμβολοσειρές

3. Οι συναρτήσεις gets και puts

- Όταν εισάγουμε μία συμβολοσειρά με scanf, διαβάζεται και αποθηκεύεται μέχρι τον χαρακτήρα αλλαγής γραμμής που εισάγουμε ή μέχρι το πρώτο κενό που εισάγουμε.
- Προκειμένου να αποθηκεύονται και τα κενά που τυχόν γράφει ο χρήστης, χρησιμοποιούμε την συνάρτηση gets που έχει πρωτότυπο:

```
char *gets(char *string)
```

- Η συνάρτηση αυτή αποθηκεύει στη συμβολοσειρά string όλην την συμβολοσειρά που διαβάζεται από τον χρήστη, με τα κενά να συμπεριλαμβάνονται. Είναι ορισμένη στην βιβλιοθήκη:

```
stdio.h
```

- Στην ίδια βιβλιοθήκη ορίζεται η συνάρτηση puts που τυπώνει στην οθόνη την συμβολοσειρά που δέχεται ως όρισμα ακολουθούμενη από ένα '\n':

```
int puts(char *string)
```

- Που επίσης έχει οριστεί στη βιβλιοθήκη

```
stdio.h
```

B. Συμβολοσειρές

4. Η βιβλιοθήκη string.h

- Είδαμε μία πρώτη εισαγωγή στις συμβολοσειρές.
- Επειδή η διαχείριση συμβολοσειρών είναι μια συνηθισμένη διαδικασία στην C, έχει οριστεί μία βιβλιοθήκη, η:

```
string.h
```

- η οποία ορίζει συναρτήσεις που κάνουν πιο εύκολη την επεξεργασία συμβολοσειρών.
- Εκεί ορίζονται συναρτήσεις όπως η συνάρτηση:

```
void strcpy(char *dest, char *src)
```

- Η οποία αντιγράφει την συμβολοσειρά src στην συμβολοσειρά dest.
- Και η συνάρτηση:

```
int strlen(char *string)
```

- Που επιστρέφει το πλήθος των χαρακτήρων της συμβολοσειράς string.
- Θα δούμε αναλυτικά την βιβλιοθήκη αυτή σε επόμενο μάθημα

Γ. Ασκήσεις

1. Μήκος Συμβολοσειράς

- Το πλήθος των συμβόλων μιας συμβολοσειράς (εκτός του ειδικού χαρακτήρα \0) αναφέρεται σαν το μήκος της συμβολοσειράς.
- Κατασκευάστε μία συνάρτηση με πρωτότυπο `int mystrlen(char *s)` που δέχεται ως όρισμα μία συμβολοσειρά και επιστρέφει το μήκος της.
- Γράψτε ένα πρόγραμμα C, το οποίο θα διαβάζει από την είσοδο μια συμβολοσειρά με χρήση της `gets` και έπειτα θα υπολογίζει και θα τυπώνει το μήκος της συμβολοσειράς

- Η συνάρτηση `int strlen(char *s)` που έχει οριστεί στο `string.h` εκτελεί ακριβώς την ενέργεια που περιγράφεται παραπάνω.

Γ. Ασκήσεις

2. Αντιγραφή Συμβολοσειρών

- Κατασκευάστε μία συνάρτηση με πρωτότυπο `char *mystrcpy(char *dest, char *src)` που δέχεται ως ορίσματα δύο συμβολοσειρές και αντιγράφει τη συμβολοσειρά `src` στην συμβολοσειρά `dest`. Η συνάρτηση να επιστρέφει έναν δείκτη στην συμβολοσειρά `dest`.
- Γράψτε ένα πρόγραμμα C, το οποίο θα διαβάζει από την είσοδο δύο συμβολοσειρές με χρήση της `gets` και έπειτα θα δίνει μία επιλογή στο χρήστη για να αντιγράψει όποια από τις δύο συμβολοσειρές επιθυμεί σε μία τρίτη συμβολοσειρά. Τελικά να τυπώνει και τις τρεις συμβολοσειρές στην οθόνη.

- Η συνάρτηση `char *strcpy(char *dest, char *src)` που έχει οριστεί στο `string.h` εκτελεί ακριβώς την ενέργεια που περιγράφεται παραπάνω.

Γ. Ασκήσεις

3. Μετατροπή σε Κεφαλαία

- Γράψτε ένα πρόγραμμα που να διαβάζει μια συμβολοσειρά από τον χρήστη και να μετατρέπει κάθε μικρό γράμμα στο αντίστοιχο κεφαλαίο και να τυπώνει το αποτέλεσμα στην οθόνη
- Υποδείξεις:
 - Δηλώστε μία αρκετά μεγάλη συμβολοσειρά (π.χ. 150 χαρακτήρων)
 - Παρατηρήστε από τον πίνακα χαρακτήρων ASCII ότι κάθε μικρός χαρακτήρας διαφέρει από τον αντίστοιχο κεφαλαίο κατά 32 ακριβώς θέσεις.
 - Σκεφθείτε ότι θα αλλάζουν μόνο οι χαρακτήρες που είναι μικροί. Συνεπώς συμβουλευθείτε τον πίνακα ASCII για να δείτε ποιοι χαρακτήρες είναι οι μικροί.
 - Αποφασίστε αν θα χρησιμοποιήσετε την `gets` ή την `scanf` και για ποιο λόγο.

Γ. Ασκήσεις

4. Μορφοποίηση Εξόδου

- Δείτε ότι στο 2^ο πακέτο χαρακτήρων του πίνακα ASCII (128-255) υπάρχουν κάποιοι χαρακτήρες που με διπλή γραμμή μπορούν να απεικονίσουν ένα πλαίσιο.
- Χρησιμοποιήστε τους χαρακτήρες αυτούς για να εκτυπώσετε το μήνυμα (μαζί με το πλαίσιο):

```
Hello World!
```



Γ. Ασκήσεις

5. Σύγκριση συμβολοσειρών

- Κατασκευάστε πρόγραμμα C το οποίο:
 - Να διαβάζει δύο λέξεις με μικρούς λατινικούς χαρακτήρες (να πραγματοποιηθεί έλεγχος ότι ο χρήστης πληκτρολόγησε μικρούς λατινικούς χαρακτήρες)
 - Να πραγματοποιεί αλφαβητική σύγκριση των συμβολοσειρών και να τυπώνει κατάλληλο μήνυμα.
- Παράδειγμα επιθυμητής εκτέλεσης:

```
Dwse tin 1i simvoloseira: test
Dwse tin 2i simvoloseira: abba
=====
Isxyei: abba < test
```

- Παρόμοια ενέργεια επιτελεί η συνάρτηση `int strcmp(char *s1, char *s2)` της βιβλιοθήκης `string.h` την οποία θα μελετήσουμε σε επόμενο μάθημα