

# ΕΠΛ 033: ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ

Μάριος Belk, Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

Email: [belk@cs.ucy.ac.cy](mailto:belk@cs.ucy.ac.cy)



# ΣΥΜΒΟΛΟΣΕΙΡΕΣ

# Πέρασμα πινάκων σε Συναρτήσεις

2

- Πρότυπα Συναρτήσεων που δέχονται τιμές διεύθυνσης προς πίνακα

```
void modifyArray( int b[], int arraySize);
```

- Για να μπορέσουμε να χρησιμοποιήσουμε σωστά τη διεύθυνση που λαμβάνουμε, χρειάζεται να ξέρουμε σε τι τύπο δεδομένων δείχνει και πόσα στοιχεία αυτού του τύπου ακολουθούν (μέγεθος πίνακα)

# Συμβολοσειρές

3

- Οι **Συμβολοσειρές** είναι **Πίνακες Χαρακτήρων** που περιέχουν το **σήμα τέλους** `'\0'`
  - Οι Συμβολοσειρές έχουν ιδιαίτερη μεταχείριση στη C:
- Μια **συμβολοσειρά** μπορεί να αρχικοποιηθεί και χρησιμοποιώντας τα **σύμβολα** `" "`

```
char string1[] = "first";
```

- είναι ισοδύναμο ουσιαστικά με τον πιο κάτω ορισμό :

```
char string1[] = { 'f', 'i', 'r', 's', 't', '\0' };
```

- Υπάρχει ειδικός χαρακτήρας (`'\0'`) για να καθορίζει το **τέλος ενός string**
- Η C εκμεταλλεύεται αυτό το σήμα τέλους για να μπορεί να τυπώσει όλα τα στοιχεία του πίνακα χρησιμοποιώντας απλά την `printf()` και την ειδική παράμετρο `"%s"`

Για να τυπώσουμε το περιεχόμενο του `string1`

```
printf("You are the %s", string1);
```

# Συμβολοσειρές

4

- Οι Συμβολοσειρές είναι Πίνακες Χαρακτήρων που περιέχουν το σήμα τέλους `'\0'`
  - Οι Συμβολοσειρές έχουν ιδιαίτερη μεταχείριση στη C:
- Με παρόμοιο τρόπο η C επιτρέπει στον χρήστη να αναθέσει τιμές σε μια συμβολοσειρά άμεσα χρησιμοποιώντας απλά την `scanf()` ως εξής:

```
printf("Please enter your name: ");  
scanf("%s", string2);
```

- Η `scanf()` θα διαβάσει όλους τους χαρακτήρες που θα δώσει ο χρήστης μέχρι το πρώτο `space`
- Αν θέλουμε να διαβάσουμε όλους τους χαρακτήρες που δίνει ο χρήστης μέχρι να πατήσει `enter`, πχ. αν θέλουμε να μπουν στη συμβολοσειρά οι χαρακτήρες `"John Smith"`, τότε θα χρησιμοποιούμε την παράμετρο `"%[^\n]s"` στην `scanf()` αντί για το απλό `"%s"`

# Συμβολοσειρές

5

Ένας πίνακας χαρακτήρων μπορεί να αρχικοποιηθεί χρησιμοποιώντας τα σύμβολα " " .

```
char string1[] = "first";
```

Ο Null χαρακτήρας '\0' καθορίζει το τέλος ενός strings

Το μέγεθος του string `string1` είναι 6, είναι ισοδύναμο με τον πιο κάτω ορισμό :

```
char string1[] = { 'f', 'i', 'r', 's', 't', '\0' };
```

```
char string1[]="first";
```

```
char string1[6]="first";
```

```
char string2[20];
```

→ Είναι δυνατό να έχουμε πρόσβαση σε καθένα από τα στοιχεία του (χαρακτήρες) ξεχωριστά.

`string1[ 3 ]` είναι ουσιαστικά ο χαρακτήρας 's'

# Συναρτήσεις και Πίνακες Χαρακτήρων

6

```
void copy1(char s1[], char s2[])
{
    int i = 0;
    do {
        s2[i]=s1[i];
        i++;
    } while ( s1[i]!='\0' )
    return;
}
```

Μία με μία οι τιμές μεταφέρονται από τον ένα πίνακα στον άλλο χρησιμοποιώντας το index του πίνακα

```
void copy1(char *s1, char *s2)
{
    do {
        *s2=*s1;
        s2++;
        s1++;
    } while ( *s1!='\0' )
    return;
}
```

Μία με μία οι τιμές μεταφέρονται από τον ένα πίνακα στον άλλο χρησιμοποιώντας αυξάνοντας το που δείχνει ο κάθε δείκτης στη μνήμη

# #include <string.h>

1 7	<b>strcpy(s1, s2);</b> Copies string s2 into string s1.
2	<b>strcat(s1, s2);</b> Concatenates string s2 onto the end of string s1.
3	<b>strlen(s1);</b> Returns the length of string s1.
4	<b>strcmp(s1, s2);</b> Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
5	<b>strchr(s1, ch);</b> Returns a pointer to the first occurrence of character ch in string s1.
6	<b>strstr(s1, s2);</b> Returns a pointer to the first occurrence of string s2 in string s1.



# Παράδειγμα strcpy, strcat, strlen

8

```
#include <stdio.h>
#include <string.h>
int main ()
{
char str1[12] = "Hello";
char str2[12] = "World";
char str3[12];
int len ;
/* copy str1 into str3 */
strcpy(str3, str1);
printf("strcpy( str3, str1) : %s\n", str3 );

/* concatenates str1 and str2 */
strcat( str1, str2);
printf("strcat( str1, str2): %s\n", str1 );

/* total length of str1 after concatenation */
len = strlen(str1);
printf("strlen(str1) : %d\n", len );
return 0;
}
```

C:\Users\admin\Dropbox\EPL033-2015-2016\Lectures\17\strings\_example.exe

```
strcpy( str3, str1) : Hello  
strcat( str1, str2): HelloWorld  
strlen(str1) : 10  
Press any key to continue . . . _
```

# Παράδειγμα strcmp

10

```
#include <stdio.h>
#include <string.h>
int main ()
{
char str1[15];
char str2[15];
int ret;
strcpy(str1, "abcdef");
strcpy(str2, "ABCDEF");
ret = strcmp(str1, str2);
if(ret < 0)
    { printf("str1 is less than str2"); }
else if(ret > 0)
    { printf("str2 is less than str1"); }
else
    { printf("str1 is equal to str2"); }
return(0); }
```

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	SOH (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	STX (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	ETX (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	EOT (end of transmission)	36	24	044	&#36;	\$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	ENQ (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	ACK (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	BEL (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	BS (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	VT (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	CR (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	SO (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	SI (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	DLE (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	DC1 (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	DC2 (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	DC3 (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	DC4 (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	CAN (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	EM (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	SUB (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	ESC (escape)	59	3B	073	&#59;	:	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	FS (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	
29	1D	035	GS (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	RS (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	US (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Source: [www.LookupTables.com](http://www.LookupTables.com)

str2 is less than str1

# Παράδειγμα - strcpy

3-11

- Ζητούμενο: να γράψετε συνάρτηση που αντιγράφει ένα αλφαριθμητικό  $t$  στο αλφαριθμητικό  $s$ .
- Γιατί δεν είναι αρκετό να γράψουμε  $s=t$ ;
- **Εκδοχή με πίνακες:**

```
void strcpy (char *s, char *t) {  
    int i;  
    i=0;  
    while (t[i] != '\0') {  
        s[i] = t[i];  
        i++;  
    }  
}
```

# Παράδειγμα - strcpy

3-12

- Εκδοχή με δείκτες:

```
void strcpy1 (char *s, char *t)
{
    while (*t != '\0') {
        *s = *t;
        s++;
        t++;
    }
}
```

# Παράδειγμα 2 - strcmp

3-13

- Ζητούμενο: γράψετε συνάρτηση που συγκρίνει τα αλφαριθμητικά  $s_1$  και  $s_2$  και επιστρέφει «1» αν το  $s_1$  μεγαλύτερο λεξικογραφικά, “0” αν είναι ίσα και «-1» αν  $s_2$  μεγαλύτερο λεξικογραφικά

	0	1	2	3	4	5	6	7	8	9
$s_1$	C	U	T	\0	?	?	?	?	?	?

>

	0	1	2	3	4	5	6	7	8	9
$s_2$	C	A	T	\0	?	?	?	?	?	?

Επιστρέφει 1

# Παράδειγμα 2 - strcmp

## A) Εκδοχή με πίνακες (from P.J. Plauger, Standard C Library)

```
int mystrcmp(char s1[], char s2[]) {
    int i;

    for(i=0; s1[i] == s2[i]; i++)
        if(s1[i] == 0) // i.e., NULL
            return 0;

    if (s1[i] < s2[i]) return -1;
    else return 1;
}
```

# Παράδειγμα 2 - strcmp

## B) Εκδοχή με δείκτες

```
int mystrcmp(char *s1, char *s2) {  
  
    for(; *s1 == *s2; ++s1, ++s2)  
        if(*s1 == 0)  
            return 0;  
  
    if (*s1 < *s2) return -1;  
    else return 1;  
}
```



# Συμβολοσειρές

16

Function Name	Function Purpose
<code>isalnum</code>	Check if character is alphanumeric
<code>isalpha</code>	Check if character is alphabetic
<code>isblank</code>	Check if character is blank
<code>iscntrl</code>	Check if character is a control character
<code>isdigit</code>	Check if character is decimal digit
<code>isgraph</code>	Check if character has graphical representation
<code>islower</code>	Check if character is lowercase letter
<code>isprint</code>	Check if character is printable
<code>ispunct</code>	Check if character is a punctuation character
<code>isspace</code>	Check if character is a white-space
<code>isupper</code>	Check if character is uppercase letter
<code>isxdigit</code>	Check if character is hexadecimal digit