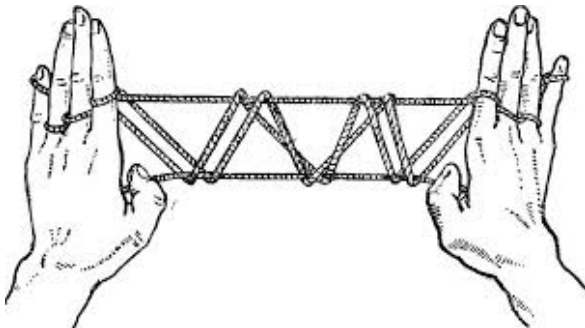


Δομημένος προγραμματισμός

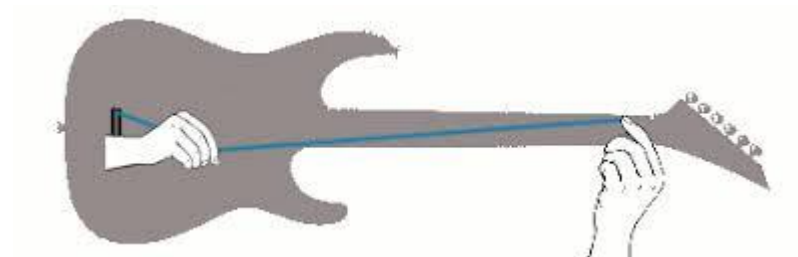


<https://www.geograph.org.uk>

Συμβολοσειρές,
χειρισμός χαρακτήρων

Περίγραμμα της ύλης που θα καλυφθεί

- Οι συμβολοσειρές στη C
- Τρόπος αποθήκευσης
- Ανάθεση τιμών σε συμβολοσειρές
- Διάβασμα / εμφάνιση συμβολοσειρών
- Συναρτήσεις με συμβολοσειρές



<https://commons.wikimedia.org/>

Οι συμβολοσειρές (strings) στη C

- Συμβολοσειρά ονομάζουμε μια οποιαδήποτε ακολουθία αλφαριθμητικών χαρακτήρων:

«Κώστας», «Κώστας Βασιλάκης», «Δαιδάλου 23»

- Στην C δεν υπάρχει ιδιαίτερος τύπος για τον χειρισμό των συμβολοσειρών.
- Για την αποθήκευση τους χρησιμοποιούμε πίνακες τύπου *char* και φυσικά τις διαχειριζόμαστε όπως τους πίνακες:

char line[80]; // δήλωση συμβολοσειράς με 80 χαρακτήρες

line[i] = getchar(); //απόδοση τιμής σε στοιχείο της σειράς

- Αν πρέπει λοιπόν να διαχειριστούμε *N* χαρακτήρες, ο πίνακας της συμβολοσειράς θα πρέπει να δηλωθεί με *N+1* χαρακτήρες.
- Υπάρχουν ειδικές συναρτήσεις για τη διαχείριση των συμβολοσειρών (βιβλιοθήκη *<string.h>*)

Παρατηρήσεις για την αποθήκευση συμβολοσειρών

- Ο τελευταίος χαρακτήρας μιας συμβολοσειράς είναι ο `'\0'` (μηδενικός χαρακτήρας – *null character*).
- Ο μηδενικός χαρακτήρας `'\0'` είναι διαφορετικός από τον χαρακτήρα `'0'` (μηδέν). Ο μηδενικός χαρακτήρας έχει την τιμή 0 (εσωτερική αναπαράσταση: 00000000), ενώ ο χαρακτήρας `'0'` έχει την τιμή 48 (00110000). Δηλαδή η εντολή:

`printf(" %d, %d", '\0','0');` // θα δώσει σαν αποτέλεσμα: 0, 48

- Οι τιμές που δίνουμε στις μεταβλητές που έχουν δηλωθεί σαν συμβολοσειρές, οριοθετούνται με λατινικά εισαγωγικά διπλά (") ή απλά ('). Προσοχή το `'A'` δεν είναι το ίδιο με το `"A"` :

Το `'A'` χρειάζεται 1 Byte μνήμης (01000001)

Το `"A"` χρειάζεται 2 Bytes (`'A','\0'` ==> 0100000100000000)

- Το `"A"` είναι μια συμβολοσειρά που περιέχει το χαρακτήρα (γράμμα) `'A'` και στο τέλος τον χαρακτήρα `'\0'` .

Αρχικές τιμές σε συμβολοσειρές (με τη δήλωση)

- Όπως στους πίνακες:

```
char str1[6 ] = {'h', 'e', 'l', 'l', 'o', '\0'};
```

(To str1[0] είναι h, to str1[1] είναι e και το str1[5] είναι '\0')

- Εναλλακτικά:

```
char str2[8] = "hello"; // Το '\0' μπαίνει αυτόματα στο τέλος
```

- Επίσης,

```
char str2[] = "hello"; //το πλήθος υπολογίζεται αυτόματα
```

- Αν δώσουμε: `char str[20] = "hello";` δεσμεύουμε 20 θέσεις μνήμης, ενώ χρειαζόμαστε μόνο 6. Οι θέσεις που περισσεύουν παίρνουν τη τιμή '\0'.
- Η εντολή `char str[20] = {'\0'};` δίνει την τιμή '\0' σε όλα τα στοιχεία της συμβολοσειράς `str`.
- Η `sizeof(str)` επιστρέφει τον αριθμό χαρακτήρων της σειράς.

Παράδειγμα με αρχικές τιμές

- Όλες οι δηλώσεις που ακολουθούν:

(α) `char CC[] = "abcd";`

είτε,

(β) `char CC[50] = "abcd";`

είτε,

(γ) `char CC[] = {'a', 'b', 'c', 'd', '\0'};`

είτε,

(δ) `char CC[5] = {'a', 'b', 'c', 'd', '\0'};`

έχουν σαν αποτέλεσμα το ίδιο περιεχόμενο.

Όμως, ενώ για τις (α), (γ), και (δ) η CC θα έχει μέγεθος 5 Bytes, στη (β) η CC θα έχει μέγεθος 50 Bytes.

Διάβασμα από το πληκτρολόγιο – scanf(), gets()

- Με την γνωστή scanf() με χρήση του προσδιοριστή %s:

```
printf("What's your name :"); scanf("%s", str);
```

- Προσοχή: δεν βάζουμε τον χαρακτήρα & μπροστά από το str διότι το str είναι η διεύθυνση του 1ου στοιχείου της συμβολοσειράς.
 - Επίσης προσοχή: Η scanf() σταματά τη ανάγνωση μόλις συναντήσει τον κενό χαρακτήρα (space) ή το χαρακτήρα ελέγχου για τη αλλαγή γραμμής ('\n').
- Εναλλακτικά, για το διάβασμα μιας συμβολοσειράς από το πληκτρολόγιο μπορούμε να χρησιμοποιήσουμε τη συνάρτηση gets():
 - *printf("What's your name:"); gets(name);*
 - Η gets() διαβάζει συνεχώς από το πληκτρολόγιο μέχρι να συναντήσει τον χαρακτήρα '\n' (ακόμα και τους κενούς χαρακτήρες). Αντικαθιστά δε τον '\n' με τον '\0'.

Εμφάνιση συμβολοσειρών στη οθόνη

- Με την συνάρτηση `printf()` και τον προσδιοριστή `%s`:

```
printf("To onoma sou einai: %s\n", onoma);
```

Η `printf()` εμφανίζει όλους τους χαρακτήρες μέχρι να συναντήσει το μηδενικό χαρακτήρα `'\0'`.

- Με την `printf()` μπορούμε να εμφανίσουμε μέρος της συμβολοσειράς:

```
printf("To onoma sou einai: %s\n", onoma+2);
```

(εμφανίζει την συμβολοσειρά από τον 3ο χαρακτήρα και μετά)

- Με τη συνάρτηση `puts()`:

```
printf("What's your name:"); gets(name);
```

```
printf("Hello "); puts(name);
```

Η `puts()` μετά την εμφάνιση της συμβολοσειράς, στέλνει αυτόματα το χαρακτήρα `'\n'`.

Τρεις διαφορετικοί τρόποι να διαβάσουμε μια συμβολοσειρά

```
#include <stdio.h>

main() {    // με scanf()
    char line [20];
    printf("Enter line: ");
    scanf("%s", line);
    printf("Line: %s\n", line);
}
```

```
#include <stdio.h>

main() {    // με gets()
    char line[20];
    printf("Enter line: ");
    gets(line);
    printf("Line: ");
    puts(line);
}
```

```
#include <stdio.h>

main() {    // με την getchar();
    char line[30], ch;
    int i = 0;
    printf("Enter line: ");
    while(ch != '\n') { //μέχρι το newline
        ch = getchar();
        line[i] = ch;
        i++;
    }
    line[i] = '\0'; //για να γίνει συμβολοσειρά
    printf("Line: %s", line);
}
```

Άλλοι τρόποι με το getchar()

```
int i;  
char line[80];  
i = 0;  
//διαβάζει χαρακτήρες μέχρι να διαβαστεί ο '\n'  
while( (line[i]=getchar( )) != '\n' ) i++;  
//δεν πρέπει να ξεχνάμε το '\0' στο τέλος  
line[i]='\0';  
printf("Diavasa %d xaraktires\n", i);
```

*Αν οι χαρακτήρες είναι
περισσότεροι από 80;*

```
//Το ίδιο με άλλο τρόπο  
while( (line[i++]=getchar( )) != '\n' );  
line[i] = '\0';
```

Συναρτήσεις της <string.h>

Συνάρτηση	Περιγραφή
<i>strcpy(s1, s2)</i>	Αντιγράφει το <i>s2</i> στο <i>s1</i>
<i>strcat(s1, s2)</i>	Συνενώνει τη <i>s2</i> στο τέλος της <i>s1</i>
<i>strlen(s1)</i>	Επιστρέφει το μήκος του <i>s1</i> (σε χαρακτήρες)
<i>strcmp(s1, s2);</i>	Επιστρέφει 0 αν τα <i>s1</i> και <i>s2</i> είναι ίδια. Μικρότερο του 0 αν <i>s1</i> < <i>s2</i> . Μεγαλύτερο του 0 αν <i>s1</i> > <i>s2</i> .
<i>strchr(s1, ch);</i>	Επιστρέφει τον δείκτη της πρώτης εμφάνισης του χαρακτήρα <i>ch</i> στη συμβολοσειρά <i>s1</i> .
<i>strstr(s1, s2);</i>	Επιστρέφει το δείκτη της πρώτης εμφάνισης της συμβολοσειράς <i>s2</i> στη συμβολοσειρά <i>s1</i> .

Παραδείγματα με συναρτήσεις από τη <string.h>

```
#include <stdio.h>
#include <string.h>
main () {
    char str1[12] = "Hello";
    char str2[12] = "World";
    char str3[12];
    int len ;
    strcpy(str3, str1);    // αντιγράφει str1 στο str3
    printf("strcpy( str3, str1) : %s\n", str3 );
    strcat( str1, str2);    // ενώνει str1 και str2
    printf("strcat( str1, str2): %s\n", str1 );
    len = strlen(str1);    // το μήκος του str1 μετά τη συνένωση
    printf("strlen(str1) : %d\n", len );
}
```

Συναρτήσεις της <ctype.h>

Συνάρτηση	Περιγραφή
<i>int isalnum(int C)</i>	Αληθές αν C είναι αλφαριθμητικό
<i>int isalpha(int C)</i>	Αληθές αν C είναι γράμμα
<i>int isascii(int C)</i>	Αληθές αν C είναι χαρακτήρας ASCII.
<i>int iscntrl(int C)</i>	Αληθές αν C είναι χαρακτήρας ελέγχου.
<i>int isdigit(int C)</i>	Αληθές αν C είναι αριθμός.
<i>int isgraph(int C)</i>	Αληθές αν C είναι χαρακτήρας γραφικών
<i>int islower(int C)</i>	Αληθές αν C είναι χαρακτήρας (αλφαβήτου) πεζός
<i>int isprint(int C)</i>	Αληθές αν C είναι χαρακτήρας που εμφανίζεται στην οθόνη
<i>int ispunct (int C)</i>	Αληθές αν C είναι χαρακτήρας στίξης
<i>int isspace(int C)</i>	Αληθές αν C είναι το κενό (space)
<i>int isupper(int C)</i>	Αληθές αν C είναι χαρακτήρας (αλφαβήτου) κεφαλαίος
<i>int isxdigit(int C)</i>	Αληθές αν C είναι δεκαεξαδικό ψηφίο

Παραδείγματα με συναρτήσεις από τη <ctype.h>

```
#include<stdio.h>
#include<ctype.h>
int main() {
    char ch;
    printf("Please enter a character: ");
    scanf("%c",&ch);
    if (isupper(ch)) printf("%c is an uppercase character.\n",ch);
    if (islower(ch)) printf("%c is a lowercase character.\n",ch);
    if (isdigit(ch))  printf("%c is a digit\n",ch);
    if (ispunct(ch))  printf("%c is a punctuation character\n",ch);
    if (isspace(ch))  printf("%c is space.\n",ch);
    return 0;
}
```

Και κάποιες ακόμα συναρτήσεις

- Από τη `<ctype.h>` (μετατροπές χαρακτήρων)

Συνάρτηση	Περιγραφή
<i>int toascii(int A)</i>	Μετατρέπει τον A σε χαρακτήρα ASCII
<i>int tolower(int C)</i>	Μετατρέπει τον C σε χαρακτήρα πεζό
<i>int toupper(int C)</i>	Μετατρέπει τον C σε χαρακτήρα κεφαλαίο

- Και από την `<stdlib.h>`

Συνάρτηση	Περιγραφή
<code>double atof(const char *str)</code>	Μετατροπή συμβολοσειράς σε float
<code>int atoi(const char *str)</code>	Μετατροπή συμβολοσειράς σε integer
<code>long int atol(const char *str)</code>	Μετατροπή συμβολοσειράς σε long

Παραδείγματα με συναρτήσεις από τη <stdlib.h>

```
int main() {  
    int ival;  
    double dval;  
    char str[30];  
    strcpy(str, "9899"); //συμβολοσειρά στο str  
    ival = atoi(str); //μετατροπή της συμβολοσειράς σε ακέραιο  
    printf("String value: %s, as integer: %d\n", str, ival);  
    strcpy(str, "9899.34"); //συμβολοσειρά στο str  
    dval = atof(str); //μετατροπή της συμβολοσειράς σε float  
    printf("String value: %s, as float: %f\n", str, dval);  
    strcpy(str, "Electrical Engineering"); //συμβολοσειρά στο str  
    ival = atoi(str); //??????  
    printf("String value:%s, as integer: %d\n", str, ival);  
    dval = atof(str); //??????  
    printf("String value = %s, as float: %f\n", str, dval);  
    return(0);  
}
```

Παραδείγματα με συναρτήσεις από τη <stdlib.h>

```
String value: 9899, as integer: 9899
String value: 9899.34, as float: 9899.340000
String value: Electrical engineering, as integer: 0
String value: Electrical engineering, as float: 0.000000
```

```
strcpy(str, "9899"); //συμβολοσειρά στο str
ival = atoi(str); //μετατροπή της συμβολοσειράς σε ακέραιο
printf("String value: %s, as integer: %d\n", str, ival);
strcpy(str, "9899.34"); //συμβολοσειρά στο str
dval = atof(str); //μετατροπή της συμβολοσειράς σε float
printf("String value: %s, as float: %f\n", str, dval);
strcpy(str, "Electrical Engineering"); //συμβολοσειρά στο str
ival = atoi(str); //??????
printf("String value:%s, as integer: %d\n", str, ival);
dval = atof(str); //??????
printf("String value = %s, as float: %f\n", str, dval);
return(0);
}
```

Και κάποια άλλα παραδείγματα

```
char string7[] = "abc";  
int len = strlen(string7);  
printf("%d\n", len);
```

```
char string5[20] = "Hello, ";  
char string6[] = "world!";  
printf("%s\n", string5);  
strcat(string5, string6);  
printf("%s\n", string5);
```

```
char string8[] = "this is a test";  
char string9[10];  
substr(string9, string8, 5, 4);  
printf("%s\n", string9)
```

```
char string3[] = "this is";  
char string4[] = "a test";  
if(strcmp(string3, string4) == 0)  
    printf("strings are equal\n");  
else    printf("strings are different\n");
```

```
for(i = 0; src[i] != '\0'; i++)  
    dest[i] = src[i];  
dest[i] = '\0';
```

```
for(i = 0; ; i++) {  
    dest[i] = src[i];  
    if(src[i] == '\0') break;  
}
```

```
char string[] = "123";  
int i = atoi(string);  
int j = atoi("456");
```