

Η ΓΛΩΣΣΑ C

Μάθημα 18:

Η Βιβλιοθήκη string.h

Δημήτρης Ψούνης



www.psounis.gr



Περιεχόμενα Μαθήματος

A. Η Βιβλιοθήκη string.h

1. Εισαγωγή
2. Βασικές Συναρτήσεις του string.h
 1. Μήκος Συμβολοσειράς (strlen)
 2. Αντιγραφή Συμβολοσειράς (strcpy)
 3. Συνένωση Συμβολοσειράς (strcat)
 4. Σύγκριση Συμβολοσειρών (strcmp)
 5. Αναζήτηση Συμβολοσειράς (strstr)
3. Περισσότερες Συναρτήσεις του string.h
 1. Η συνάρτηση strncpy
 2. Η συνάρτηση strncat
 3. Η συνάρτηση strncmp
 4. Η συνάρτηση strdup
 5. Η συνάρτηση strtok
4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά
 1. Η συνάρτηση strchr
 2. Η συνάρτηση strrchr
 3. Η συνάρτηση strcspn
 4. Η συνάρτηση strspn
5. Επιπλέον Συναρτήσεις του string.h
 1. Συναρτήσεις που δεν ορίζονται στο ANSI πρότυπο
 2. Συναρτήσεις ελέγχου χαρακτήρων

B. Ασκήσεις



A. Η Βιβλιοθήκη string.h

1. Εισαγωγή

- Οι συμβολοσειρές είναι σημαντικό μέρος σχεδόν σε οποιοδήποτε πρόγραμμα C.
- Στο «Μάθημα 10: Χαρακτήρες και Συμβολοσειρές» είδαμε το συντακτικό των συμβολοσειρών και κατασκευάσαμε κάποιες συναρτήσεις για τον χειρισμό συμβολοσειρών.
- Στο σημερινό μάθημα, θα μελετήσουμε την βιβλιοθήκη συναρτήσεων string.h η οποία υλοποιεί ένα σύνολο από συναρτήσεις που είναι πολύ πιθανό να είναι χρήσιμες στα προγράμματά μας:
 - Για αντιγραφή και συνένωση συμβολοσειρών
 - Για σύγκριση συμβολοσειρών
 - Για αναζήτηση σε συμβολοσειρές
 - Και αρκετές ακόμη!

Σημείωση:

- Το μάθημα αυτό είναι το πρώτο που αφιερώνουμε εξ' ολοκλήρου σε μια βιβλιοθήκη συναρτήσεων της C.
- Η C έχει αρκετές βιβλιοθήκες με πολύ χρήσιμες συναρτήσεις. Θα μελετήσουμε και άλλες βιβλιοθήκες σε επόμενα μαθήματα.



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

- Οι πιο συχνά χρησιμοποιούμενες συναρτήσεις του string.h είναι οι ακόλουθες πέντε:

Όνομα	Πρωτότυπο Συνάρτησης	Λειτουργία
strlen	<code>size_t strlen(char *str)</code>	Επιστρέφει το μήκος της συμβολοσειράς
strcpy	<code>char *strcpy(char *dest, const char *src)</code>	Αντιγράφει στη συμβολοσειρά dest τη συμβολοσειρά src
strcat	<code>char *strcat(char *str1, const char *str2)</code>	Συνένωση συμβολοσειρών (str1=str1+str2)
strcmp	<code>int strcmp(const char *str1, const char *str2)</code>	Σύγκριση συμβολοσειρών (Επιστρέφει <0 αν str1<str2, =0 αν str1=str2, >0 αν str1>str2)
strstr	<code>char *strstr(const char *str, const char *search)</code>	Επιστρέφει δείκτη στην 1 ^η εμφάνιση της search στην str

- Εκτελούν τις βασικές πράξεις που είναι συχνότερο να είναι χρήσιμες σε ένα πρόγραμμα χειρισμού συμβολοσειρών.



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

1. Μήκος Συμβολοσειράς (strlen)

- Το πρωτότυπο της συνάρτησης strlen:

```
size_t strlen(char *str)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται σαν όρισμα μία συμβολοσειρά:
 - Επιστρέφει το μήκος της (πλήθος χαρακτήρων) που περιέχει
 - Δεν μετρείται το '\0'.

Σημείωση:

- Το `size_t` είναι ένα (περίπου) «συνώνυμο» του τύπου δεδομένων `unsigned int`
- Όπου το συναντάμε θα αντιμετωπίζουμε το επιστρεφόμενο αποτέλεσμα ως ακέραιο αριθμό



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

1. Μήκος Συμβολοσειράς (strlen)

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strlen

```
/* strlen.c: Epideikniei ti xrisi tis strlen */  
  
#include <stdio.h>  
#include <string.h>  
  
#define N 80  
  
main()  
{  
    char str[N];  
  
    /* eisodos */  
    printf("Dwse ti simvoloseira: ");  
    gets(str);  
  
    /* eksodos */  
    printf("Mikos=%d", strlen(str));  
}
```



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

2. Αντιγραφή Συμβολοσειρών (strcpy)

- Το πρωτότυπο της συνάρτησης strcpy:

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

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται σαν όρισμα δύο συμβολοσειρές:
 - Αντιγράφει την συμβολοσειρά src στην συμβολοσειρά dest (περιλαμβάνοντας το χαρακτήρα \0)
 - Επιστρέφει ένα δείκτη στη συμβολοσειρά dest.

Σημείωση:

- Είναι ευθύνη του προγραμματιστή να έχει δεσμεύσει χώρο και αυτός να είναι επαρκής για τη συμβολοσειρά dest
- Η strcpy είναι υποχρεωτική όταν θέλουμε να αντιγράψουμε συμβολοσειρές. Η εντολή `str1=str2` αναθέτει τον δείκτη str2 να δείχνει στην μνήμη που δείχνει η str1 άρα ΔΕΝ κάνει αντιγραφή συμβολοσειράς.



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

2. Αντιγραφή Συμβολοσειρών (strcpy)

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strcpy

```
/* strcpy.c: Epideikniei ti xrisi tis strcpy */
#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    char str1[N], str2[N];

    /* eisodos */
    printf("Dwse ti simvoloseira: ");
    gets(str1);

    /* antigrafi me tin strcpy */
    strcpy(str2, str1);

    /* eksodos */
    printf("str1=%s, str2=%s", str1, str2);
}
```




A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

3. Συνένωση Συμβολοσειρών (strcat)

- Το πρωτότυπο της συνάρτησης strcat:

```
char *strcat(char *str1, const char *str2)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται σαν όρισμα δύο συμβολοσειρές str1 και str2 και:
 - Αντιγράφει στο τέλος της str1 ένα αντίγραφο της str2
 - Βάζει στο τέλος της νέας str1 ένα \0
 - Επιστρέφει ένα δείκτη στη συμβολοσειρά str1.

ΠΡΙΝ:

```
str1 Hello
```

```
str2 World
```

META:

```
str1 HelloWorld
```

```
str2 World
```

Σημείωση:

- Είναι ευθύνη του προγραμματιστή να έχει δεσμεύσει χώρο και αυτός να είναι επαρκής για τη συμβολοσειρά που θα προκύψει από τη συνένωση των συμβολοσειρών.



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

3. Συνένωση Συμβολοσειρών (strcat)

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strcat

```
/* strcat.c: Epideikniei ti xrisi tis strcat */

#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    char str1[N], str2[N];

    /* eisodos */
    printf("Dwse ti simvoloseira 1: ");
    gets(str1);
    printf("Dwse ti simvoloseira 2: ");
    gets(str2);

    /* antigrafi me tin strcat */
    strcat(str1, str2);
    printf("\nstr1=%s", str1);
    strcat(str1, str2);
    printf("\nstr1=%s", str1);
}
```



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

4. Σύγκριση Συμβολοσειρών (strcmp)

- Το πρωτότυπο της συνάρτησης strcmp:

```
int strcmp(const char *str1, const char *str2)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται σαν όρισμα δύο συμβολοσειρές και ανάλογα με την αλφαριθμητική σύγκρισή τους:
 - Αν $str1 < str2$ επιστρέφει ακέραιο < 0
 - Αν $str1 == str2$ επιστρέφει 0
 - Αν $str1 > str2$ επιστρέφει ακέραιο > 0



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

4. Σύγκριση Συμβολοσειρών (strcmp)

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strcmp

```
/* strcmp.c: Anadeiknuei ti xrisi tis strcmp */
#include <stdio.h>
#include <string.h>
#define N 80

main()
{
    int res;
    char str1[N], str2[N];

    /* eisodos */
    printf("Dwse ti simvoloseira 1: ");
    gets(str1);
    printf("Dwse ti simvoloseira 2: ");
    gets(str2);

    /* sygrisi me tin strcmp */
    res=strcmp(str1,str2);
    if (res<0)
        printf("str1<str2 (res=%d)",res);
    else if (res==0)
        printf("str1==str2 (res=%d)",res);
    else
        printf("str1>str2 (res=%d)",res);
}
```



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

5. Αναζήτηση Συμβολοσειράς (strstr)

- Το πρωτότυπο της συνάρτησης strstr:

```
char *strstr(const char *str, const char *search)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται μία συμβολοσειρά (str)
- Επιστρέφει:
 - Δείκτη στην 1^η εμφάνιση της search στην str.
 - NULL αν δεν υπάρχει η search στην str



A. Η Βιβλιοθήκη string.h

2. Βασικές Συναρτήσεις του string.h

5. Αναζήτηση Συμβολοσειράς (strstr)

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strstr

```
/* strstr.c: Anadeiknuei ti xrisi tis strstr */
#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    char str1[N], str2[N];
    char *ptr;

    /* eisodos */
    printf("Dwse to keimeno sto opoio anazitoume: ");
    gets(str1);
    printf("Dwse ti anazitoume: ");
    gets(str2);

    ptr=strstr(str1,str2);

    while(ptr!=NULL)
    {
        printf("Stin thesi %d: %s\n",ptr-str1,ptr);
        ptr=strstr(ptr+1,str2);
    }
}
```



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

- Οι ακόλουθες συναρτήσεις χρησιμοποιούνται επίσης συχνά στην πράξη. Είναι συναρτήσεις:
 - Που συγκεκριμενοποιούν το μήκος συμβολοσειράς στις συνήθεις πράξεις (strncpy, strncat, strncmp)
 - Διευκόλυνσης χρήσης της αντιγραφής (strdup)
 - Διαχωρισμού συμβολοσειράς σε λέξεις (strtok)

Όνομα	Πρωτότυπο Συνάρτησης	Λειτουργία
strncpy	<code>char *strncpy(char *dest, const char *src, size_t n)</code>	Αντιγράφει η χαρακτήρες από τη συμβολοσειρά dest στη συμβολοσειρά src
strncat	<code>char *strncat(char *str1, const char *str2, size_t n)</code>	Συνένωση συμβολοσειρών (str1=str1+str2) με συγκεκριμένο μήκος της str2.
strncmp	<code>int strncmp(const char *str1, const char *str2, size_t n)</code>	Σύγκριση των n πρώτων χαρακτήρων συμβολοσειρών (Επιστρέφει <0 αν str1<str2, =0 αν str1=str2, >0 αν str1>str2)
strdup	<code>char *strdup(const char *str)</code>	Επιστρέφει ένα αντίγραφο της συμβολοσειράς str (δεσμεύοντας μνήμη για αυτήν)
strtok	<code>char *strtok(char *str, char *delim)</code>	Διαχωρίζει την συμβολοσειρά str με βάση τα διαχωριστικά της delim



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

1. Η συνάρτηση strncpy()

- Το πρωτότυπο της συνάρτησης strncpy:

```
char *strncpy( char *dest, const char *src, size_t n)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται σαν όρισμα δύο συμβολοσειρές και έναν ακέραιο:
 - Αν το μήκος της `src` είναι μικρότερο του `n`:
 - Αντιγράφει τη συμβολοσειρά `src` στην συμβολοσειρά `dest` (περιλαμβάνοντας το χαρακτήρα `\0`)
 - Αν το μήκος της `src` είναι μεγαλύτερο(ή ίσο) του `n`:
 - Αντιγράφει τους πρώτους `n` χαρακτήρες από την συμβολοσειρά `src` στην συμβολοσειρά `dest` (χωρίς να προσθέτει το χαρακτήρα `\0`)
 - Επιστρέφει ένα δείκτη στη συμβολοσειρά `dest`.



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

1. Η συνάρτηση strncpy()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strcpy

```
/* strncpy.c: Epideikniei ti xrisi tis strncpy */  
  
#include <stdio.h>  
#include <string.h>  
  
#define N 80  
  
main()  
{  
    char str1[N], str2[N];  
    int l=5;  
  
    /* eisodos */  
    printf("Dwse ti simvoloseira: ");  
    gets(str1);  
  
    /* antigrafi me tin strncpy */  
    strncpy(str2, str1, l);  
  
    /* eksodos */  
    printf("str1=%s, str2=%s", str1, str2);  
}
```



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

2. Η συνάρτηση strncat()

- Το πρωτότυπο της συνάρτησης strncat:

```
char *strncat(char *str1, const char *str2, size_t n)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται σαν όρισμα δύο συμβολοσειρές και έναν ακέραιο:
 - Αν το μήκος της str2 είναι μικρότερο του n:
 - Αντιγράφει στο τέλος της str1 ένα αντίγραφο της str2.
 - Βάζει στο τέλος της νέας str1 ένα \0
 - Αν το μήκος της str2 είναι μεγαλύτερο(ή ίσο) του n:
 - Αντιγράφει στο τέλος της str1 τους πρώτους n χαρακτήρες της str2.
 - Βάζει στο τέλος της νέας str1 το \0
- Επιστρέφει ένα δείκτη στη συμβολοσειρά str1.



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

2. Η συνάρτηση strncat()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strncat

```
/* strncat.c: Epideikniei ti xrisi tis strncat */  
  
#include <stdio.h>  
#include <string.h>  
  
#define N 80  
  
main()  
{  
    int i;  
    char str1[N], str2[N];  
  
    /* eisodos */  
    printf("Dwse ti simvoloseira 1: ");  
    gets(str1);  
    printf("Dwse ti simvoloseira 2: ");  
    gets(str2);  
  
    /* antigrafi me tin strncat */  
    for (i=1; i<=5; i++)  
    {  
        strncat(str1, str2, i);  
        printf("\nstr1=%s", str1);  
    }  
}
```



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

3. Η συνάρτηση strncmp()

- Το πρωτότυπο της συνάρτησης strncmp:

```
int strncmp(const char *str1, const char *str2, size_t n)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται σαν όρισμα δύο συμβολοσειρές και έναν ακέραιο n και συγκρίνει τους πρώτους n χαρακτήρες (ή μέχρι το \0):
 - Αν `str1 < str2` επιστρέφει ακέραιο `< 0`
 - Αν `str1 == str2` επιστρέφει `0`
 - Αν `str1 > str2` επιστρέφει ακέραιο `> 0`



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

3. Η συνάρτηση strncmp()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strncmp

```
/* strncmp.c: Anadeiknuei ti xrisi tis strncmp */
#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    int res,l=4;
    char str1[N], str2[N];

    /* eisodos */
    printf("Dwse ti simvoloseira 1: ");
    gets(str1);
    printf("Dwse ti simvoloseira 2: ");
    gets(str2);
    /* sygkrisi me tin strncmp */
    res=strncmp(str1,str2,l);
    /* eksodos */
    if (res<0)
        printf("str1<str2 (res=%d)",res);
    else if (res==0)
        printf("str1==str2 (res=%d)",res);
    else
        printf("str1>str2 (res=%d)",res);
}
```



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

4. Η συνάρτηση strdup()

- Το πρωτότυπο της συνάρτησης strdup:

```
char *strdup(const char *str)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Προσοχή ότι δεν είναι ANSI (ενδέχεται να μην υποστηρίζεται από τον μεταγλωττιστή)
- Η συνάρτηση δέχεται ως όρισμα μία συμβολοσειρά.
- Δεσμεύει κατάλληλο χώρο (με malloc), αντιγράφει τη συμβολοσειρά στο δεσμευμένο χώρο και επιστρέφει:
 - Έναν δείκτη στη νέα συμβολοσειρά.
 - NULL αν δεν έγινε η αντιγραφή



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

4. Η συνάρτηση strdup()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strdup

```
/* strdup.c: Anadeiknuei ti xrisi tis strdup */
#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    char str[N];
    char *str2;

    /* eisodos */
    printf("Dwse ti simvoloseira: ");
    gets(str);
    /* antigrافي me tin strdup */
    str2=strdup(str);
    /* eksodos */
    if (!str2)
    {
        printf("Adynamia desmeusis mnimis!");
    }
    else{
        printf("Antigrafike: %s me mikos: %d",str2,strlen(str2));
    }
}
```



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

5. Η συνάρτηση strtok()

- Το πρωτότυπο της συνάρτησης strtok:

```
char *strtok(char *str, char *delim)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται ως όρισμα μία συμβολοσειρά str και μία συμβολοσειρά που κωδικοποιεί διαχωριστικά (π.χ. κόμμα, τελεια,κενό: “,. “).
- Αν το str δεν είναι NULL (πρώτη κλήση), τότε επιστρέφει έναν δείκτη στην αρχή της συμβολοσειράς, έχοντας αντικαταστήσει την πρώτη εμφάνιση διαχωριστικού με \0.
- Αν το str είναι NULL (επόμενες κλήσεις) τότε συνεχίζει η λειτουργία της από το σημείο που η τελευταία κλήση βρήκε έναν από τους χαρακτήρες διαχωρισμού.

Σημείωση:

- Το παράδειγμα της επόμενης διαφάνειας αναδεικνύει τη συνάρτηση.



A. Η Βιβλιοθήκη string.h

3. Περισσότερες Συναρτήσεις του string.h

5. Η συνάρτηση strtok()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strtok

```
/* strtok.c: Anadeiknuei ti xrisi tis strtok */
#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    int i,res;
    char str[N], *p;

    /* eisodos */
    printf("Dwse ti simvoloseira: ");
    gets(str);
    /* xwrismos me tin strtok */
    p=strtok(str, " ,.");
    /* eksodos */
    while(p!=NULL)
    {
        printf("%s\n",p);
        p=strtok(NULL, " ,.");
    }
}
```



A. Η Βιβλιοθήκη string.h

4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά

- Ως συνέχεια της συνάρτησης strstr υπάρχουν και οι εξής συναρτήσεις που κάνουν αναζήτηση συμβολοσειράς

Όνομα	Πρωτότυπο Συνάρτησης	Λειτουργία
strchr	<code>char *strchr(const char *str, int ch)</code>	Επιστρέφει δείκτη στην 1 ^η εμφάνιση του χαρακτήρα <code>ch</code> στην <code>str</code>
strrchr	<code>char *strrchr(const char *str, int ch)</code>	Επιστρέφει δείκτη στην 1 ^η εμφάνιση του χαρακτήρα <code>ch</code> στην <code>str</code> (αντιστρόφως)
strcspn	<code>size_t strcspn(const char *str, const char *search)</code>	Επιστρέφει τη πρώτη θέση που βρέθηκε χαρακτήρας της <code>search</code> στην <code>str</code>
strspn	<code>size_t strspn(const char *str, const char *search)</code>	Επιστρέφει πόσοι χαρακτήρες της <code>search</code> βρέθηκαν στην αρχή της <code>str</code>



A. Η Βιβλιοθήκη string.h

4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά

1. Η συνάρτηση strchr()

- Το πρωτότυπο της συνάρτησης strchr:

```
char *strchr(const char *str, int ch)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται ως ορίσματα μία συμβολοσειρά και έναν χαρακτήρα.
- Επιστρέφει έναν δείκτη στην πρώτη εμφάνιση του χαρακτήρα στην συμβολοσειρά.
 - Αν ο χαρακτήρας δεν υπάρχει στην συμβολοσειρά επιστρέφει NULL.



A. Η Βιβλιοθήκη string.h

4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά

1. Η συνάρτηση strchr()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strchr

```
/* strchr.c: Anadeiknuei ti xrisi tis strchr */
#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    int i,res;
    char str[N], ch;
    char *ptr;

    /* eisodos */
    printf("Dwse ti simvoloseira: ");
    gets(str);
    printf("Dwse to xaraktira: ");
    ch=getchar();
    /* anazitisi me tin strchr */
    ptr=strchr(str,ch);
    while(ptr!=NULL)
    {
        puts(ptr);
        ptr=strchr(ptr+1,ch);
    }
}
```



A. Η Βιβλιοθήκη string.h

4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά

2. Η συνάρτηση strrchr()

- Το πρωτότυπο της συνάρτησης strrchr:

```
char *strrchr(const char *str, int ch)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται ως ορίσματα μία συμβολοσειρά και έναν χαρακτήρα.
- Επιστρέφει έναν δείκτη στην τελευταία εμφάνιση του χαρακτήρα στην συμβολοσειρά.
 - Αν ο χαρακτήρας δεν υπάρχει στην συμβολοσειρά επιστρέφει NULL.



A. Η Βιβλιοθήκη string.h

4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά

2. Η συνάρτηση strrchr()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strchr

```
/* strrchr.c: Anadeiknuei ti xrisi tis strrchr */
#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    char str[N], ch;
    char *ptr;

    /* eisodos */
    printf("Dwse ti simvoloseira: ");
    gets(str);
    printf("Dwse to xaraktira: ");
    ch=getchar();
    /* anazitisi me tin strrchr */
    ptr=strrchr(str,ch);

    puts(ptr);
}
```

A. Η Βιβλιοθήκη string.h

4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά

3. Η συνάρτηση strcspn()

- Το πρωτότυπο της συνάρτησης strcspn:

```
size_t strcspn(const char *str, const char *search)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται μία συμβολοσειρά (str) στην οποία ψάχνει για την 1^η εμφάνιση οποιουδήποτε από τους χαρακτήρες της συμβολοσειράς (search)
- Επιστρέφει:
 - Την θέση της 1^{ης} εμφάνισης (αν βρέθηκε εμφάνιση)
 - Το μήκος της συμβολοσειράς str (αν δεν βρέθηκε εμφάνιση)



A. Η Βιβλιοθήκη string.h

4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά

3. Η συνάρτηση strcspn()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strcspn

```
/* strcspn.c: Anadeiknuei ti xrisi tis strcspn */
#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    char str[N], ch;
    char *ptr;
    int pos;

    /* eisodos */
    printf("Dwse ti simvoloseira: ");
    gets(str);

    /* anazitisi me tin strcspn */
    pos=strcspn(str,"abc");
    if (pos!=strlen(str))
        printf("Apo ti thesi %d: %s",pos, &str[pos]);
    else
        printf("Den vrethikan oi xaraktires abc stin %s", str);
}
```




A. Η Βιβλιοθήκη string.h

4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά

4. Η συνάρτηση strstr()

- Το πρωτότυπο της συνάρτησης strstr:

```
size_t strstr(const char *str, const char *search)
```

- Και έχει οριστεί στη βιβλιοθήκη: `string.h`
- Η συνάρτηση δέχεται μία συμβολοσειρά (str)
- Σε αυτήν μετράει πόσοι συνεχόμενοι (από την αρχή) χαρακτήρες της str περιέχονται στην συμβολοσειρά search
 - και επιστρέφει το πλήθος τους.



A. Η Βιβλιοθήκη string.h

4. Συναρτήσεις Αναζήτησης σε Συμβολοσειρά

4. Η συνάρτηση strstr()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση strstr

```
/* strstr.c: Anadeiknyei ti xrisi tis strstr */
#include <stdio.h>
#include <string.h>

#define N 80

main()
{
    char str1[N], str2[N];
    char *ptr;
    int i,length;

    /* eisodos */
    printf("Dwse ti 1i simvoloseira: ");
    gets(str1);
    printf("Dwse ti 2i simvoloseira: ");
    gets(str2);

    /* anazitisi me tin strstr */
    length=strstr(str1,str2);
    if (length>0){
        printf("Oi prwtoi %d xaraktires tis str1 periexontai stin stin
                str2:",length);
        for (i=0; i<length; i++)
            printf("%c",str1[i]);
    }
    else
        printf("Oi symvoloseires den tairiazoun");
}
```



A. Η Βιβλιοθήκη string.h

5. Επιπλέον Συναρτήσεις του string.h

1. Συναρτήσεις που δεν ορίζονται στο ANSI πρότυπο

- Οι ακόλουθες συναρτήσεις δεν ορίζονται στο πρότυπο ANSI, αλλά έχουν υλοποιηθεί στους περισσότερους μεταγλωττιστές:

Όνομα	Λειτουργία	Πρωτότυπο Συνάρτησης	Λειτουργία
strlwr	Μετατροπή σε μικρά	<code>char *strlwr(char *str)</code>	Δέχεται ως όρισμα μια συμβολοσειρά Μετατρέπει τους κεφαλαίους χαρακτήρες σε μικρούς Επιστρέφει δείκτη στην <code>str</code>
strupr	Μετατροπή σε κεφαλαία	<code>char *strupr(char *str)</code>	Δέχεται ως όρισμα μια συμβολοσειρά Μετατρέπει τους μικρούς χαρακτήρες σε κεφαλαίους Επιστρέφει δείκτη στην <code>str</code>
strrev	Αντιστροφή Συμβολοσειράς	<code>char *strrev(char *str)</code>	Δέχεται ως όρισμα μια συμβολοσειρά Αντιστρέφει τη συμβολοσειρά Επιστρέφει δείκτη στην <code>str</code>



A. Η Βιβλιοθήκη string.h

5. Επιπλέον Συναρτήσεις του string.h

1. Συναρτήσεις που δεν ορίζονται στο ANSI πρότυπο

- Οι ακόλουθες συναρτήσεις δεν ορίζονται στο πρότυπο ANSI, αλλά έχουν υλοποιηθεί στους περισσότερους μεταγλωττιστές:

Όνομα	Λειτουργία	Πρωτότυπο Συνάρτησης	Λειτουργία
strset	Αντικατάσταση χαρακτήρων συμβολοσειράς	<code>char *strset(char *str, int ch)</code>	Δέχεται ως όρισμα μια συμβολοσειρά. Γράφει σε όλη τη συμβολοσειρά το χαρακτήρα <code>ch</code> . Επιστρέφει δείκτη στην <code>str</code>
strnset	Αντικατάσταση χαρακτήρων συμβολοσειράς	<code>char *strnset(char *str, int ch, size_t n)</code>	Δέχεται ως όρισμα μια συμβολοσειρά. Αντικαθιστά τους <code>n</code> πρώτους χαρακτήρες της <code>str</code> με <code>ch</code> . Επιστρέφει δείκτη στην <code>str</code>



A. Η Βιβλιοθήκη string.h

5. Επιπλέον Συναρτήσεις του string.h

2. Συναρτήσεις ελέγχου χαρακτήρων

- Οι ακόλουθες συναρτήσεις χρησιμεύουν για τον έλεγχο ενός χαρακτήρα:

Όνομα	Πρωτότυπο Συνάρτησης	Λειτουργία
isalnum	<code>int isalnum(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι αλφαριθμητικό (χαρακτήρας ή ψηφίο)
isalpha	<code>int isalpha(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι γράμμα
isdigit	<code>int isdigit(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι ψηφίο
ispunct	<code>int ispunct(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι σημείο στίξης
isctrl	<code>int isctrl(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι το CTRL
isgraph	<code>int isgraph(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι εκτυπώσιμος χαρακτήρας (εκτός του space)
isprint	<code>int isprint(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι εκτυπώσιμος χαρακτήρας (περιλαμβάνει το space)



A. Η Βιβλιοθήκη string.h

5. Επιπλέον Συναρτήσεις του string.h

2. Συναρτήσεις ελέγχου χαρακτήρων

- Οι ακόλουθες συναρτήσεις χρησιμεύουν για τον έλεγχο ενός χαρακτήρα:

Όνομα	Πρωτότυπο Συνάρτησης	Λειτουργία
isupper	<code>int isupper(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι κεφαλαίος
islower	<code>int islower(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι μικρός
isspace	<code>int isspace(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι κενό (space, tab, vertical tab, form feed, ή carriage return)
isblank	<code>int isblank(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι κενό
isxdigit	<code>int isxdigit(char ch)</code>	Επιστρέφει TRUE/FALSE αν ο <code>ch</code> είναι 16δικό ψηφίο



B. Ασκήσεις

1. Ταξινόμηση Συμβολοσειρών

Στο μάθημα «Αλγόριθμοι σε C: Μάθημα 3 – Ταξινόμηση Πίνακα» κατασκευάσαμε ένα πρόγραμμα που υλοποιεί ταξινόμηση ακεραίων (π.χ. με την bubblesort). Τροποποιήστε το πρόγραμμα έτσι ώστε:

1. Να διαβάζει από την είσοδο 10 συμβολοσειρές
2. Να ταξινομεί τις συμβολοσειρές χρησιμοποιώντας για την σύγκριση τη συνάρτηση strcmp.
3. Να προβάλλει τον ταξινομημένο πίνακα στην οθόνη.



B. Ασκήσεις

2. Ταξινόμηση Συμβολοσειρών

Τροποποιήστε το πρόγραμμα της εφαρμογής 1, έτσι ώστε αντί να διαβάζει 10 συμβολοσειρές, να διαβάζει μία φράση, να την αποδομεί σε λέξεις με την strtok και έπειτα να ταξινομεί τις λέξεις που προέκυψαν.