

Η ΓΛΩΣΣΑ C

Μάθημα 15:

Είσοδος/Εξοδος: Συναρτήσεις Εισόδου

Δημήτρης Ψούνης



Περιεχόμενα Μαθήματος

A. Ροές

1. Γενικά
2. Η ροή προκαθορισμένης εισόδου `stdin`
3. Η ροή προκαθορισμένης εξόδου `stdout`
4. Η ροή προκαθορισμένης εξόδου λαθών `stderr`

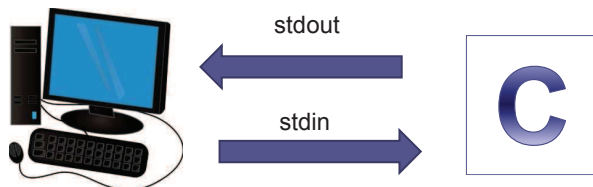
B. Συναρτήσεις ροής εισόδου `stdin`

1. Γενικά
2. Η συνάρτηση `getchar()`
3. Η συνάρτηση `getch()`
4. Η συνάρτηση `getche()`
5. Η συνάρτηση `ungetc()`
6. Η συνάρτηση `gets()`
7. Η συνάρτηση `fgets()`
8. Η συνάρτηση `scanf()`
9. Η συνάρτηση `fflush()`

A. Ροές

1. Γενικά

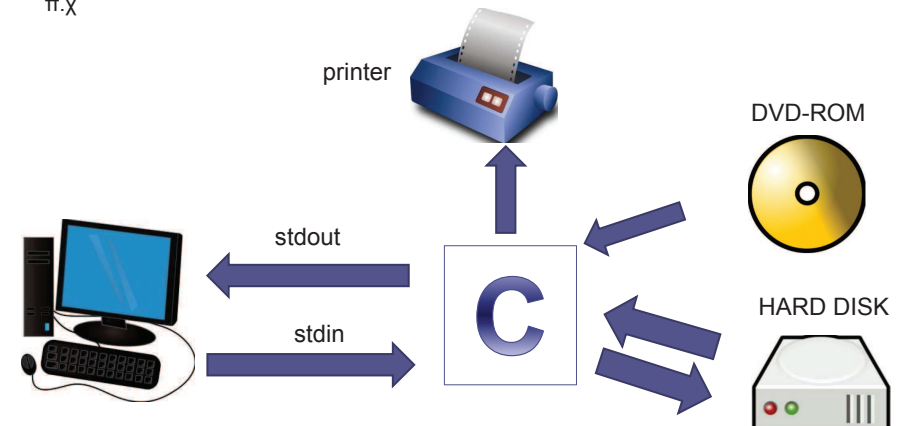
- Η είσοδος και η έξοδος στην C, γίνεται μέσω ροών (streams).
- Μία ροή είναι μια ακολουθία χαρακτήρων η οποία είναι εν αναμονή για να την χρησιμοποιήσουμε ανάλογα με την θέλησή μας.
- Ήδη (χωρίς να το γνωρίζουμε) έχουμε χρησιμοποιήσει δύο ροές:
 - Την προκαθορισμένη ροή εισόδου (`stdin`) από την οποία διαβάζουμε χαρακτήρες από το πληκτρολόγιο.
 - Την προκαθορισμένη ροή εξόδου (`stdout`) στην οποία γράφουμε χαρακτήρες στην οθόνη.



A. Ροές

1. Γενικά

- Κάθε ροή της C λοιπόν είναι μια ακολουθία χαρακτήρων την οποία μπορούμε να διαχειριστούμε από το πρόγραμμά μας. Πέρα από τις 2 σημαντικές ροές που έχουμε δουλέψει έχουν οριστεί πολλές ακόμη ροές που μπορούν να διαχειριστούν είσοδο/έξοδο από και προς άλλες συσκευές, π.χ



A. Ροές

2. Η ροή προκαθορισμένης εισόδου stdin

- Η ANSI C (διεθνές πρότυπο με τις βασικές λειτουργίες που πρέπει να έχει κάθε μεταγλωττιστής της C) ορίζει ότι οι παρακάτω 5 ροές υπάρχουν σε κάθε υλοποίηση της C:

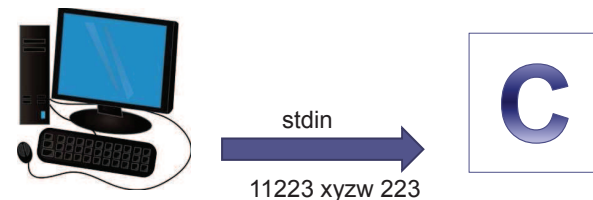
Όνομα	Ροή	Συσκευή
stdin	Προκαθορισμένη Είσοδος	Πληκτρολόγιο
stdout	Προκαθορισμένη Έξοδος	Οθόνη
stderr	Προκαθορισμένο Λάθος	Οθόνη
stdprn*	Προκαθορισμένος Εκτυπωτής	Εκτυπωτής (LPT1)
stdaux*	Προκαθορισμένο Βοήθημα	Σειριακή Θύρα (COM1)

- Οι ροές με * χρησιμοποιούνταν σε παλιότερες εκδόσεις του DOS και θεωρούνται πλέον παρωχημένες.

A. Ροές

2. Η ροή προκαθορισμένης εισόδου stdin

- Καλό είναι να σκεφτόμαστε την stdin σαν μια «σωλήνα» η οποία συνδέει το πληκτρολόγιο με το πρόγραμμά μας

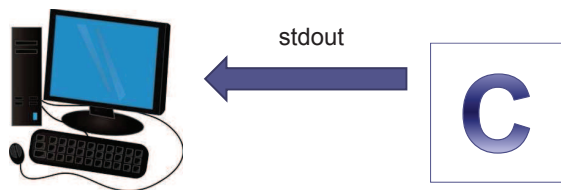


- Π.χ. Η συνάρτηση scanf που μάθαμε, διαβάζει τόσους χαρακτήρες από την stdin, όσοι καθορίζονται από τον προσδιοριστή που έχουμε θέσει σαν 1ο όρισμα. Αν ο χρήστης έχει εισάγει περισσότερους χαρακτήρες τότε αυτοί παραμένουν στην stdin για επόμενο διάβασμα από κάποια εντολή διαχείρισης της προκαθορισμένης εισόδου.
- Θα δούμε και άλλες συναρτήσεις που διαχειρίζονται την προκαθορισμένη είσοδο.

A. Ροές

3. Η ροή προκαθορισμένης εισόδου stdout

- Αντίστοιχα η stdout είναι μία «σωλήνα» που συνδέει το πρόγραμμά μας με την οθόνη του υπολογιστή.

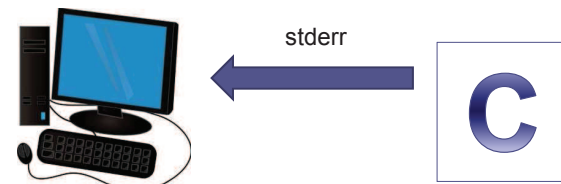


- Π.χ. Η συνάρτηση puts, τοποθετεί στην stdout την συμβολοσειρά που δέχεται ως όρισμα και δίνει εντολή να τυπωθούν οι χαρακτήρες που βρίσκονται στην stdout στην οθόνη.

A. Ροές

4. Η ροή προκαθορισμένης εξόδου λαθών stderr.

- Η stderr είναι μια αυτόνομη ροή που λειτουργεί όπως η stdout, στην οποία γράφουμε μηνύματα λάθους εκτέλεσης του προγράμματός μας.



- Σε μεγάλα προγράμματα είναι συχνό να σταματάνε να λειτουργούν για λόγους που δεν είναι απαραίτητα προγραμματιστικά λάθη (π.χ. Αδυναμία δέσμευσης μνήμης, αποτυχία υλικού, ασύγχρονης επικοινωνίας κ.λπ.)
- Προκειμένου να μην μπερδευτούν τα μηνύματα που θέλουμε να προβάλλουμε στην οθόνη, με τα σημαντικά μηνύματα αστοχίας του προγράμματος, η C ορίζει αυτήν την βοηθητική ροή, για να γίνει διαχωρισμός της εξόδου προγράμματος, από τα μηνύματα λάθους.



Β. Συναρτήσεις Ροής Εισόδου

1. Σύνοψη

- Η C μας παρέχει μία σειρά από συναρτήσεις για να διαχειριστούμε την είσοδο από το πληκτρολόγιο.
- Δύο είναι τα βασικά χαρακτηριστικά των συναρτήσεων που πρέπει να έχουμε υπόψη μας:
 - Αν οι χαρακτήρες που πληκτρολογούμε μένουν σε έναν προσωρινό χώρο αποθήκευσης και πηγαίνουν στην stdin πατώντας Enter, ή αν κάθε χαρακτήρας που πληκτρολογούμε πηγαίνει κατευθείαν στην stdin. Το χαρακτηριστικό αυτό θα το ονομάσουμε ενδιάμεση αποθήκευση χαρακτήρων
 - Αν ότι πληκτρολογούμε προβάλλεται ταυτόχρονα και στην οθόνη ή όχι. Το χαρακτηριστικό αυτό θα το ονομάσουμε προβολή χαρακτήρων



Β. Συναρτήσεις Ροής Εισόδου

1. Σύνοψη

- Ας ρίξουμε μια πρώτη ματιά στις συναρτήσεις που θα μελετήσουμε στις επόμενες διαφάνειες:

Όνομα	Λειτουργία	Ενδιάμεση Αποθήκευση Χαρακτήρων	Προβολή Χαρακτήρων	Βιβλιοθήκη
getchar	Διάβασμα ενός χαρακτήρα εισόδου	NAI	NAI	stdio.h
getch	Διάβασμα ενός χαρακτήρα εισόδου	OXI	OXI	conio.h
getche	Διάβασμα ενός χαρακτήρα εισόδου	OXI	N AΙ	conio.h
ungetc	Αναίρεση λήψης χαρακτήρα	-	-	stdio.h
gets	Διάβασμα Συμβολοσειράς	NAI	NAI	stdio.h
fgets	Διάβασμα Συμβολοσειράς	NAI	NAI	stdio.h
scanf	Διάβασμα Συμβολοσειράς	NAI	NAI	stdio.h



Β. Συναρτήσεις Ροής Εισόδου

2. Η συνάρτηση getchar()

- Το πρωτότυπο της συνάρτησης getchar είναι:

```
int getchar();
```

- Και έχει οριστεί στη βιβλιοθήκη: `stdio.h`
- Η συνάρτηση διαβάζει έναν χαρακτήρα από την stdin και επιστρέφει τον ASCII κωδικό της.
 - Λειτουργεί με ενδιάμεση αποθήκευση χαρακτήρων, άρα αναμένει από εμάς να πληκτρολογήσουμε Enter για να καταχωρηθούν οι χαρακτήρες στην stdin.
 - Προβάλλει τους χαρακτήρες που πληκτρολογούμε στην οθόνη.



Β. Συναρτήσεις Ροής Εισόδου

2. Η συνάρτηση getchar()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση getchar

```
/* getchar.c: Epideiksi tis entolis getchar */
#include <stdio.h>

main()
{
    char c;

    c=getchar();
    while(c!='x')
    {
        printf("%c",c);
        c=getchar();
    }
    printf("%c.",c);
}
```

- Ο χρήστης εισάγει μία συμβολοσειρά και στην συνέχεια πατάει Enter.
- Η είσοδος αποθηκεύεται στην stdin και περνάει στο πρόγραμμα μόλις ο χρήστης πατήσει Enter.



B. Συναρτήσεις Ροής Εισόδου

3. Η συνάρτηση getch ()

- Το πρωτότυπο της συνάρτησης getch είναι:

```
int getch ()
```

- Και έχει οριστεί στη βιβλιοθήκη: `conio.h`
- Η συνάρτηση διαβάζει έναν χαρακτήρα από την stdin και επιστρέφει τον ASCII κωδικό της.
 - Λειτουργεί χωρίς ενδιάμεση αποθήκευση χαρακτήρων, άρα με το που πατάμε έναν χαρακτήρα αυτός αποθηκεύεται στην stdin.
 - Δεν προβάλλει τους χαρακτήρες που πληκτρολογούμε στην οθόνη!!
- Προσοχή! Το αρχείο κεφαλίδας `conio.h` δεν είναι μέρος της ANSI C, άρα δεν είναι απαραίτητο να υπάρχει σε οποιονδήποτε μεταγλωττιστή. Ωστόσο το DEV-C++ την ενσωματώνει, άρα μπορούμε να την χρησιμοποιήσουμε κανονικά!



B. Συναρτήσεις Ροής Εισόδου

3. Η συνάρτηση getch ()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση getch

```
/* getch.c: Epideiksi tis entolis getch */
#include <stdio.h>
#include <conio.h>

main()
{
    char c;

    c=getch();
    while(c!='x')
    {
        printf("%c, ", c);
        c=getch();
    }
    printf("%c.", c);
}
```

- Ο χρήστης πληκτρολογεί μία συμβολοσειρά.
- Η είσοδος ΔΕΝ αποθηκεύεται στην stdin και η πληκτρολόγηση κάθε χαρακτήρα διαβάζεται αμέσως από την getch()



B. Συναρτήσεις Ροής Εισόδου

4. Η συνάρτηση getche ()

- Το πρωτότυπο της συνάρτησης getche είναι:

```
int getche ()
```

- Και έχει οριστεί στη βιβλιοθήκη: `conio.h`
- Η συνάρτηση λειτουργεί με τον ίδιο τρόπο με την getch() με την διαφορά ότι προβάλλει τον χαρακτήρα που πληκτρολογούμε στην οθόνη.
- Μεταγλωττίστε και εκτελέστε το πρόγραμμα `getche.c` που δείχνει την λειτουργία της και την διαφορά από την εντολή `getch`.



B. Συναρτήσεις Ροής Εισόδου

4. Η συνάρτηση getche ()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση getche

```
/* getche.c: Epideiksi tis entolis getch */
#include <stdio.h>
#include <conio.h>

main()
{
    char c;

    c=getche();
    while(c!='x')
    {
        printf("%c, ", c);
        c=getche();
    }
    printf("%c.", c);
}
```

- Ο χρήστης πληκτρολογεί μία συμβολοσειρά.
- Η είσοδος ΔΕΝ αποθηκεύεται στην stdin και η πληκτρολόγηση κάθε χαρακτήρα διαβάζεται αμέσως από την getche(). Όταν πληκτρολογούμε τον χαρακτήρα προβάλλεται στην οθόνη.



B. Συναρτήσεις Ροής Εισόδου

5. Η συνάρτηση ungetc()

- Το πρωτότυπο της συνάρτησης ungetc είναι :

```
int ungetc(int ch, FILE *fp)
```

- Και έχει οριστεί στη βιβλιοθήκη: `stdio.h`
- Η συνάρτηση ungetc αναιρεί την λήψη του χαρακτήρα ch από την ροή fp
 - Θα δούμε σε επόμενο μάθημα, ότι μπορούμε να καθορίσουμε και ροές εισόδου από αρχεία και γι' αυτό ως 2ο όρισμα μπορούμε να ορίσουμε και ροή αρχείου.
 - Για να αναιρέσουμε λήψη από την stdin, απλά γράφουμε ως δεύτερο όρισμα τη ροή stdin.
- Η συνάρτηση πρακτικά επανατοποθετεί στην ροή stdin τον χαρακτήρα ch που διάβασε.
- Στο παράδειγμα της επόμενης διαφάνειας βλέπουμε πως χρησιμοποιούμε την ungetc.



B. Συναρτήσεις Ροής Εισόδου

5. Η συνάρτηση ungetc()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση `ungetc`

```
/* ungetc.c: Epideiksi tis entolis ungetc */
#include <stdio.h>

main()
{
    char c;
    c=getchar();
    while(c!='\n')
    {
        printf("%c",c);
        c=getchar();
        if (c=='m')
        {
            printf("%c",c);
            ungetc('!',stdin);
        }
    }
}
```

- Πληκτρολογήστε μια συμβολοσειρά που περιέχει τουλάχιστον μία φορά το γράμμα m και δείτε τι επίδραση έχει στην stdin, λόγω του ότι επανατοποθετούμε έναν έξτρα χαρακτήρα στην stdin.



B. Συναρτήσεις Ροής Εισόδου

6. Η συνάρτηση gets()

- Το πρωτότυπο της συνάρτησης gets είναι :

```
char *gets(char *str)
```

- Και έχει οριστεί στη βιβλιοθήκη: `stdio.h`
- Έχουμε ήδη μελετήσει την συγκεκριμένη συνάρτηση σε προηγούμενο μάθημα.
- Η συνάρτηση διαβάζει μία συμβολοσειρά από την stdin μέχρι τον χαρακτήρα '\n' τον οποίο αντικαθιστά με το '\0' και την αποθηκεύει στην συμβολοσειρά str που δέχεται ως όρισμα μέσω αναφοράς.
 - Αν για κάποιο λόγο αποτύχει, επιστρέφεται η τιμή NULL (κενός δείκτης), αλλιώς επιστρέφεται δείκτης στην str.
- Ο συνηθισμένος τρόπος χρήσης της gets είναι:

```
gets(buffer);
```

 - Όπου buffer είναι συμβολοσειρά (πίνακας χαρακτήρων) που έχουμε δεσμεύσει είτε στατικά είτε δυναμικά.



B. Συναρτήσεις Ροής Εισόδου

7. Η συνάρτηση fgets()

- Πολλές φορές έχουμε πρόβλημα με την gets, διότι η συμβολοσειρά που χρησιμοποιούμε δεν έχει αρκετό χώρο για να αποθηκεύσει όλα τα δεδομένα που εισάγει ο χρήστης. Η fgets έρχεται να λύσει αυτό το πρόβλημα.
- Το πρωτότυπο της συνάρτησης fgets είναι:

```
char *fgets(char *str, int n, FILE *fp)
```

 - Και έχει οριστεί στο αρχείο βιβλιοθήκης: `stdio.h`
- Η συνάρτηση διαβάζει μία συμβολοσειρά από την ροή fp και την αποθηκεύει στην str μέχρι:
 - Είτε διαβάζει τον χαρακτήρα '\n' τον οποίο αντικαθιστά με '\0'
 - Είτε διαβάσει n-1 χαρακτήρες. Τότε προσθέτει τον χαρακτήρα '\0' στο τέλος και τερματίζει.
- Με απλά λόγια η fgets βάζει όριο στους χαρακτήρες που θα διαβάσει προκειμένου να χωράνε στην συμβολοσειρά που διοχετεύουμε ως όρισμα.
- Το παράδειγμα της επόμενης διαφάνειας αναδεικνύει την χρήση των παραπάνω



B. Συναρτήσεις Ροής Εισόδου

7. Η συνάρτηση fgets()

- Το παρακάτω παράδειγμα αναδεικνύει την συνάρτηση `fgets`

```
/* fgets.c: Epideiksi tis entolis fgets */

#include <stdio.h>

#define N 10

main()
{
    char buffer[N];

    while(1)
    {
        fgets(buffer,N,stdin);
        printf("Eisodos: %s\n",buffer);
    }
}
```

- Πληκτρολογήστε μια συμβολοσειρά με μήκος μεγαλύτερο του 10 και πατήστε Enter. Παρατηρήστε την συμπεριφορά του προγράμματος.



B. Συναρτήσεις Ροής Εισόδου

8. Η συνάρτηση scanf()

- Την συνάρτηση `scanf` την έχουμε μελετήσει ήδη δύο φορές.
- Λόγω του πλούτου που έχει η χρήση της, σήμερα θα δούμε και τις τελευταίες εξειδικευμένες χρήσεις
- Ένα τυπικό παράδειγμα χρήσης της `scanf` που έχει οριστεί στο `stdio.h` είναι το εξής:


```
scanf ("%x", &μεταβλητή);
```

 - Όπου το 1^ο όρισμα είναι η συμβολοσειρά μορφοποίησης και στα επόμενα ορίσματα ορίζουμε τις (διευθύνσεις) των μεταβλητών που θα δεχθούν τις τιμές που θα εισάγει ο χρήστης
 - Σημαντικά θέματα είναι τα εξής:
 - Τι μπορούμε να έχουμε στην συμβολοσειρά μορφοποίησης
 - Ποιοι είναι οι προσδιοριστές που μπορούμε να έχουμε στη συμβολοσειρά μορφοποίησης
 - Επίσης θα πρέπει να ξέρουμε ότι:
 - Γίνεται προβολή των χαρακτήρων στην οθόνη όσο πληκτρολογούμε.
 - Ενώ γίνεται και ενδιάμεση αποθήκευση των χαρακτήρων



B. Συναρτήσεις Ροής Εισόδου

8. Η συνάρτηση scanf()

- Στο μάθημα 9: «Εισαγωγή στην Είσοδο/Εξοδο Δεδομένων» είδαμε τους προσδιοριστές της `scanf`.
- Έχουν ωστόσο οριστεί περαιτέρω δύο ειδικές χρήσεις προσδιοριστών ώστε να καθορίζουμε ρητά τους χαρακτήρες που μπορούν να διαβαστούν (ισοδύναμα να απαγορευτεί το διάβασμα κάποιων χαρακτήρων).
- Αυτοί είναι:

Προσδιοριστής	Τύπος δεδομένων που αντιστοιχεί
[...]	Συμβολοσειρά που μόνο οι χαρακτήρες στα [] είναι αποδεκτοί. Μόλις διαβάσει χαρακτήρες εκτός των [] σταματά την ανάγνωση.
[^...]	Το ίδιο με την [], αλλά μόνο οι χαρακτήρες εκτός των [] είναι αποδεκτοί



B. Συναρτήσεις Ροής Εισόδου

8. Η συνάρτηση scanf()

- Το παρακάτω πρόγραμμα αναδεικνύει την χρήση των επιπλέον προσδιοριστών που εισάγαμε:

```
/* scanf.c: Epideiksi tis entolis scanf */

#include <stdio.h>

main()
{
    char buffer[80];

    printf("Eisagete simvoloseira: ");
    scanf ("%abcdef",buffer);
    printf("Diavastike: %s\n",buffer);
    printf("Eisagete simvoloseira: ");
    scanf ("%[^abcdef]",buffer);
    printf("Diavastike: %s",buffer);
}
```

- Πληκτρολογήστε και εκτελέστε το πρόγραμμα εισάγοντας μια φράση που περιέχει τους χαρακτήρες που παρατίθενται στον προσδιοριστή.

B. Συναρτήσεις Ροής Εισόδου

9. Η συνάρτηση fflush()

- Όταν εκτελείται μία κλήση στην scanf λαμβάνεται τόση είσοδος από την stdin όση είναι απαραίτητη για να διαβαστεί ο τύπος των δεδομένων που ορίσαμε.
- Έτσι για παράδειγμα αν έχουμε την εντολή:

```
scanf ("%d", &x) ;
scanf ("%d", &y) ;
```

- Και ο χρήστης εισάγει:

```
15 20 30
```

- Τότε διαβάζεται το 15 και ανατίθεται στη x, αγνοείται το κενό και διαβάζεται το 20 και ανατίθεται στην y. Οι υπόλοιποι χαρακτήρες παραμένουν στην stdin για επόμενο διάβασμα.
- Συνεπώς επειδή αναμένουμε πάντα κακή συμπεριφορά του χρήστη απέναντι στο πρόγραμμα πρέπει πάντα να περιμένουμε ότι ενδέχεται να έχουν απομείνει κάποιοι χαρακτήρες στην stdin μετά από κάποιο διάβασμα.
- Ειδικά ο χαρακτήρας \n απομένει στην είσοδο μετά από διάβασμα με την scanf. Για τον λόγο αυτό χρειαζόμαστε την συνάρτηση fflush.

B. Συναρτήσεις Ροής Εισόδου

9. Η συνάρτηση fflush()

- Το πρωτότυπο της συνάρτησης fflush είναι:

```
int fflush(FILE *fp)
```

- Και έχει οριστεί στο αρχείο βιβλιοθήκης: `stdio.h`
- Η συνάρτηση «αδειάζει» την stdin από τυχόν χαρακτήρες που έχουν απομείνει σε αυτήν.
- Μεταγλωττίστε το παράδειγμα της επόμενης διαφάνειας (το οποίο δεν δουλεύει σωστά) με χρήση της fflush

B. Συναρτήσεις Ροής Εισόδου

9. Η συνάρτηση fflush()

```
/* fflush_wrong.c: Epideiksi tis entolis fflush */
```

```
#include <stdio.h>
```

```
main()
{
```

```
    int x,y,z;
    char c;
```

```
    printf("Dwse enan tripsifio arithmo:");
    scanf("%ld%ld%ld", &x,&y,&z);
```

```
    printf("Epelekse M-Monades,D-Dekades,E-Ekatontades: ");
    scanf("%c",&c);
```

B. Συναρτήσεις Ροής Εισόδου

9. Η συνάρτηση fflush()

```
switch(c)
{
    case 'M':
        printf("Monades=%d",z);
        break;
    case 'D':
        printf("Dekades=%d",y);
        break;
    case 'E':
        printf("Ekatontades=%d",z);
        break;
    default:
        printf("Lathos eisodos!");
}
}
```

B. Ασκήσεις

1. Μετατροπή σε κεφαλαία (ξανά)

Γράψτε ένα πρόγραμμα το οποίο θέλουμε να:

1. Διαβάζει μία συμβολοσειρά από το πληκτρολόγιο (ΔΕΝ θέλουμε να αποθηκεύεται σε κάποιον πίνακα και ΔΕΝ θέλουμε να εμφανίζεται αυτό που πληκτρολογεί ο χρήστης)
2. Εμφανίζει στην οθόνη την ίδια συμβολοσειρά όπου κάθε μικρός χαρακτήρας έχει αντικατασταθεί από τον αντίστοιχο κεφαλαίο χαρακτήρα.

B. Ασκήσεις

2. Διάβασμα συμβολοσειρών

Γράψτε ένα πρόγραμμα το οποίο:

1. Δηλώνει έναν πίνακα χαρακτήρων (με ονομασία string) μεγέθους 10.
2. Ζητά από τον χρήστη να πληκτρολογήσει μία συμβολοσειρά.
3. Εάν η συμβολοσειρά χωράει στο string να την αποθηκεύει σε αυτό και να προβάλλει ότι «η συμβολοσειρά διαβάστηκε» και να την τυπώνει.
4. Εάν η συμβολοσειρά δεν χωράει στο string να αποθηκεύει το μέρος που μπορεί να αποθηκευτεί και να προβάλλει «η συμβολοσειρά δεν διαβάστηκε» και να τυπώνει το μέρος που «χώρεσε» στο string. Να καθαρίζει την stdin από τους χαρακτήρες που έχουν απομείνει.