

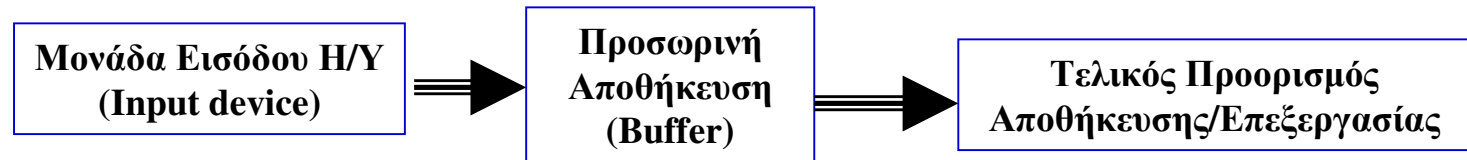
Είσοδος-Έξοδος (Input/Output)

καθιερωμένη είσοδος/έξοδος - Συναρτήσεις
Αλφαριθμητικών - Αρχεία κειμένου

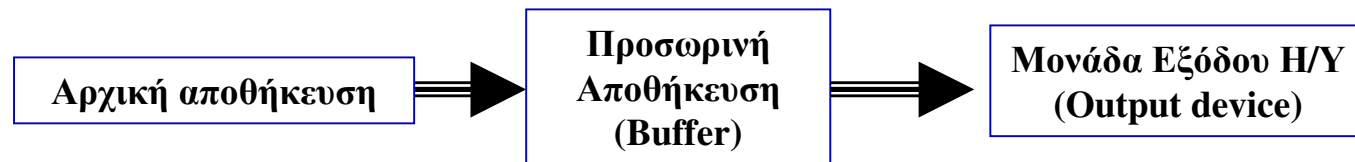
Μάθημα 5

Ροές Εισόδου/Εξόδου (streams I/O)

I-stream



O-stream



Τυπική είσοδος/έξοδος

Standard Input (*stream* **stdin**) : πληκτρολόγιο

Standard Output (*stream* **stdout**) : Κονσόλα

πάντα ανοιχτές ροές

Καθιερωμένη Βιβλιοθήκη : Επικεφαλίδα <stdio.h>

ctype.h, stdlib.h, string.h,
time.h, math.h, κλπ

Χρήση εντολής **fflush**

- `fflush(O-stream)` : Εγγραφή δεδομένων της Προσωρινής αποθήκευσης στην Έξοδο
- `fflush(I-stream)` : Απροσδιόριστο αποτέλεσμα – (πχ διαγραφή χαρακτήρα Enter)
- `fflushall()` : εκτέλεση `fflush` σε όλες τις ανοιχτές ροές

Είσοδος-Έξοδος Χαρακτήρων*

int getchar () : επιστρέφεται ο χαρακτήρας από το πληκτρολόγιο
int putchar (int) : Γράφεται ο χαρακτήρας στην κονσόλα (και επιστρέφεται από τη συνάρτηση)
Επιστρέφεται η τιμή EOF (-1) σε περίπτωση λάθους

int getc (stdio), int putc (int, stdout) : Όπως παραπάνω αλλά δηλώνεται συγκεκριμένα το όνομα της ροής προς την «τυπική» είσοδο-έξοδο.

int fgetc (stdio), int fputc (int, o-stream) : Όπως παραπάνω αλλά δηλώνεται συγκεκριμένα το όνομα της ροής από/προς αρχείο.

```
/* prog50iochar.c */
#include <stdio.h>

main()
{
    char c;
    printf("Hello, press a key : ");
    c=getchar();
    fflush(stdin);
    putchar(c);
    fflush(stdout);
    printf("\n\nend of program, press enter to exit");
    getchar();
}
```

Συναρτήσεις μετατροπής (ctype.h)

int toupper(int) : μετατροπή χαρακτήρα σε κεφαλαίο

int tolower(int) : μετατροπή χαρακτήρα σε πεζό

int isupper(int) : Επιστρέφει TRUE αν ο χαρακτήρας είναι κεφαλαίος.

Βλ. κώδικα prog51char.c

***Σημείωση :** Αν και όλες οι παραπάνω συναρτήσεις λειτουργούν με χαρακτήρες, τα ορίσματα και οι επιστροφές τους είναι τύπου int. Για επιπλέον συναρτήσεις της ctype.h δες παράρτημα B2 του βιβλίου.

Είσοδος-Έξοδος Αλφαριθμητικών

char* gets(char *) : Διαβάζει μια γραμμή εισόδου από την τυπική είσοδο και την τοποθετεί στο string char* που δηλώνεται. Βάζει αυτόματα ως τερματικό χαρακτήρα το '\0'. Επιστρέφεται ο δείκτης του string ή NULL (0) αν υπήρξε λάθος. Το <enter> (για το τέλος του string) διαγράφεται από το buffer.

int puts(char *) : Γράφει το αλφαριθμητικό στην τυπική έξοδο και αλλάζει γραμμή. Επιστρέφει τιμή -1 (EOF) σε περίπτωση λάθους ή μια μη-αρνητική τιμή.

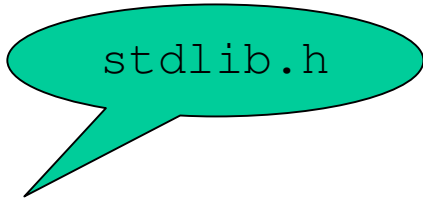
Μετατροπή αλφαριθμητικών σε αριθμούς

int atoi(char *) : μετατροπή αλφαριθμητικού σε ακέραιο

double atof(char *) : μετατροπή αλφαριθμητικού σε double

*Επιστρέφεται πάντα μηδέν αν δεν μπορεί να γίνει η μετατροπή.

char* itoa(int, *char, radix) : Μετατροπή ακέραιου αριθμού σε αλφαριθμητικό (Radix είναι η βάση μετατροπής)



stdlib.h

Βλ. κώδικα prog51str.c

Συναρτήσεις Αλφαριθμητικών (string.h)

- Θεωρούμε τα αλφαριθμητικά `char *s1, *s2`
- Προσοχή ο χώρος που έχει δεσμευτεί για τα `s1, s2` θεωρείτε επαρκής

int strlen(s1) :

Επιστρέφεται το μήκος του `s1` (δεν μετράει τον τερματικός χαρακτήρα `'\0'`).

char *strcpy(s1, s2) :

Αντιγράφει το `s2` στο `s1` (περιλαμβάνεται και ο τερματικός χαρακτήρας `'\0'`).
Επιστρέφεται ο δείκτης του `s1`

char *strcat(s1, s2) :

Συνενώνει το `s2` στο τέλος του `s1` και επιστρέφεται το `s1` (ο τερματικός χαρακτήρας `'\0'` περιέχεται στο τέλος του νέου `s1`).

strncpy(s1, s2, n) : αντιγράφει τους `n` πρώτους χαρακτήρες του `s2` στο `s1`. Αν `n` μεγαλύτερο του μήκους του `s2` τότε συμπληρώνονται οι υπόλοιποι χαρακτήρες με `'\0'` αλλιώς **δεν** μπαίνει τερματικός χαρακτήρας στο `s1`.

strncat(s1, s2, n) : συνενώνει το `s1` με τους `n` πρώτους χαρακτήρες του `s2`. Το `s1` τερματίζεται με την προσθήκη του `'\0'`.

Βλ. κώδικα prog53str1.c

Συναρτήσεις Αλφαριθμητικών (string.h)

int strcmp(s1, s2) :

Σύγκριση του αλφαριθμητικού s1 με το αλφαριθμητικό s2 .

Επιστρέφει **αρνητικό** αριθμό αν $s1 < s2$

θετικό αριθμό αν $s1 > s2$

μηδέν αν $s1 == s2$.

int strncmp(s1, s2, n) : Όπως η `strcmp` αλλά συγκρίνει μόνο τους n πρώτους χαρακτήρες των αλφαριθμητικών.

char * strchr(s, c) :

Επιστρέφεται ο δείκτης που δείχνει την 1η εμφάνιση του χαρακτήρα c μέσα στο αλφαριθμητικό s.

char * strstr(s1, s2) :

Επιστρέφεται ο δείκτης που δείχνει την 1η εμφάνιση του αλφαριθμητικού s2 μέσα στο αλφαριθμητικό s1.

Βλ. κώδικα prog53str2.c

Σημείωση : Για επιπλέον συναρτήσεις της `string.h` δεξ παράρτημα B3 του βιβλίου.

Άσκησης

Άσκηση 5.10. Γράψτε ένα κώδικα ο οποίος να αναγνωρίσει αν ένα δοθέν αλφαριθμητικό είναι αριθμός ή όχι (για πραγματικούς αριθμούς θεωρήστε μόνο την περίπτωση που ο αριθμός δίνεται σε δεκαδική μορφή με τον χαρακτήρα `.`)

Άσκηση 5.11. Γράψτε ένα πρόγραμμα στο οποίο ο χρήστης να δίνει 5 ονόματα (με μέγιστο μέγεθος 64 χαρακτήρες το καθένα) και αυτά να αποθηκεύονται σε έναν πίνακα και στη συνέχεια να εκτυπώνονται.

Άσκηση 5.12. Γράψτε ένα πρόγραμμα στο οποίο ο χρήστης να δίνει ονόματα (αλφαριθμητικά) και αυτά να αποθηκεύονται σε έναν πίνακα (χωρητικότητας N ονομάτων, πχ 100 με μέγιστο μέγεθος 64 χαρακτήρες το καθένα). Η εισαγωγή ονομάτων να σταματάει όταν ο χρήστης δώσει ως όνομα το `0`.

Άσκηση 5.13. Σε συνέχεια της άσκησης 5.11 ή 5.12, ταξινομήστε τα ονόματα που δίνονται και εκτυπώστε τα με αλφαβητική σειρά.

Ασκήσεις

Άσκηση 5.20. Γράψτε ένα κώδικα ο οποίος, με δεδομένο ένα αλφαριθμητικό "xxxx" και έναν ακέραιο N , να δημιουργεί τα αλφαριθμητικά "xxxx1.dat", "xxxx2.dat",, "xxxx N .dat". Τα αλφαριθμητικά να αποθηκεύονται σε ένα πίνακα επαρκούς χωρητικότητας.

Άσκηση 5.21. Κάντε ένα κώδικα που να δίνεται ένα όνομα αρχείου μορφής
"xxxxx.yyy"
και να μας δίνει δύο αλφαριθμητικά, το "xxxxx" (κύριο όνομα) και το "yyy" (επέκταση)

***Άσκηση 5.30.** Σχεδιάστε και υλοποιείστε το παιχνίδι «κρεμάλα».

Μορφοποιημένη Είσοδος/Εξοδος (stdio.h)

Αλφαριθμητικό μορφοποίησης (`char *format`): ένα σταθερό αλφαριθμητικό που περιέχει το ειδικό χαρακτήρα `%` και αντίστοιχους τύπους μορφοποίησης `%d`, `%f`, `%c`, `%s`, `%p` κλπ. (βλ. εντολή `printf`, μάθημα 2ο, βιβλίο σελ 220).

`int scanf(char *format, δεικτες μεταβλητών)` :

Διαβάζει χαρακτήρες από την τυπική είσοδο, τους ερμηνεύει με τη σειρά, σύμφωνα με τους τύπους μορφοποίησης του αλφαριθμητικού μορφοποίησης, και τους τοποθετεί σε θέσεις μνήμης που δηλώνονται από τους αντίστοιχους δείκτες στα υπόλοιπα ορίσματα της συνάρτησης.

Επιστρέφει τον αριθμό των αντικειμένων εισόδου.

Σταματάει όταν εισαχθούν όλα τα δεδομένα που απαιτούνται σύμφωνα με το αλφαριθμητικό μορφοποίησης ή όταν εισαχθούν δεδομένα που δεν ταιριάζουν με την μορφοποίηση.

Η `fflush(stdio)` καθαρίζει το buffer εισόδου από τυχών υπολείμματα που αφήνει η `scanf` σε περίπτωση λάθους εισαγωγής στοιχείων.

`int printf(char *format, μεταβλητές ή σταθερές)` :

Εκτυπώνει στην τυπική έξοδο ένα αλφαριθμητικό που δημιουργείτε με βάση το δοθέν αλφαριθμητικό μορφοποίησης και τα δεδομένα μεταβλητών ή σταθερών.

Επιστρέφει τον αριθμό των χαρακτήρων που εκτυπώνονται συνολικά.

Το αλφαριθμητικό μορφοποίησης μπορεί να περιέχει και χαρακτήρες στοίχισης ή πληροφορίες γηφίων (βλ. εντολή `printf`, μάθημα 2ο, βιβλίο σελ 215).

Scanf, Παραδείγματα

```
int a;  
scanf("%d", &a);
```

Διαβάζει έναν ακέραιο και τον τοποθετεί στη θέση μνήμης του ακέραιου `a`, δηλαδή δίνει ή αλλάζει τη τιμή του `a`.

```
do {  
n=scanf("%d", &a); fflush(stdin);  
} while(n!=1);
```

Διαβάζει από την τυπική είσοδο μέχρι να γίνει η σωστή εισαγωγή

```
float x; double y;  
scanf("%f %lf", &x, &y);
```

Διαβάζει δύο δεκαδικούς αριθμούς και τους τοποθετεί στις θέσεις μνήμης των `x` (απλής ακρίβειας) και `y` (διπλής ακρίβειας)

```
unsigned int a, b, c;  
scanf("%d-%d-%d", &a, &b, &c);
```

Πρέπει να εισάγουμε τρεις ακεραίους δίνοντας ανάμεσά τους το χαρακτήρα `'-'`
Πχ γράφουμε `34-56-67`.

```
ή  
int d, y; char s[4];  
scanf("%d %s %d", &d, s, &y);
```

εισάγουμε ακέραιο, αλφαριθμητικό, με τρεις χαρακτήρες το πολύ, και ακέραιο
Πχ 12 May 1975

Σημειώσεις:

- Στην μεταβλητή του αλφαριθμητικού δεν βάζουμε το σύμβολο `&` γιατί μια τέτοια μεταβλητή είναι δείκτης.
- Κατά την εισαγωγή αλφαριθμητικών να αποφεύγουμε να βάζουμε κείμενο και άλλα σύμβολα στο αλφαριθμητικό μορφοποίησης (μετά το `%s`) γιατί δεν έχουμε σωστή λειτουργία και σωστή τοποθέτηση του τερματικού `'\0'`.

Βλ. κώδικα prog55scanf.c

Ασκήσεις

Άσκηση 5.50. Κάντε ένα πρόγραμμα που να ζητάει από το χρήστη να εισάγει έναν ακέραιο αριθμό και το πρόγραμμα να βρίσκει (και να εκτυπώνει μήνυμα) αν ο ακέραιος είναι πρώτος ή όχι.

Άσκηση 5.51. Κάντε ένα πρόγραμμα που να ζητάει από το χρήστη να εισάγει μια θερμοκρασία καθώς και την μονάδα μέτρησης (χαρακτήρας C για Κελσίου ή F για Φαρεναϊτ). Το πρόγραμμα να εκτυπώνει την θερμοκρασία στην άλλη μονάδα μέτρησης. Πχ ο χρήστης εισάγει « 15 C », το πρόγραμμα να εκτυπώνει « 59 F ».
Πχ ο χρήστης εισάγει « 59 F », το πρόγραμμα να εκτυπώνει « 15 C ».
($F=32+9C/5$, $C=5(F-32)/9$)

Άσκηση 5.55. Κάντε ένα πρόγραμμα στο οποίο ο χρήστης να εισάγει την ημερομηνία γέννησης στη μορφή “dd/mm/yyyy” και το πρόγραμμα να υπολογίζει την ηλικία του (με δεκαδικό αριθμό!)

Μορφοποιημένη Είσοδος/Εξοδος από/σε Αλφαριθμητικό

int sscanf(char *s, char *format, δεικτες μεταβλητών) :

Λειτουργεί όπως η scanf (...) αλλά διαβάζει τις πληροφορίες εισόδου από ένα δεδομένο αλφαριθμητικό s .

```
char *s="15 Sep 2008";
int day, year;
char month[4];

sscanf(s, "%d %s %d", &day, month, &year);
printf("  day= %d\n", day);
printf("month= %s\n", month);
printf(" year= %d\n", year);
```

int sprintf(char *s, char *format, δεικτες μεταβλητών) :

Λειτουργεί όπως η print (...) αλλά την έξοδο, που δημιουργεί, την στέλνει στο αλφαριθμητικό s .

```
char *s[32];
int day=15, year=2008;
char month[]="Sep";

sprintf(s, "%d %s %d", day, month, year);
printf("The date is %s \n", s);
```

Αρχεία

Τα αρχεία αποτελούν σύνολα δεδομένων που είναι αποθηκευμένα σε ένα μέσο μόνιμης αποθήκευσης (πχ σκληρό δίσκο). Τα αρχεία αναγνωρίζονται από το λειτουργικό σύστημα με ένα όνομα το οποίο έχει τη δομή

Μέσο \ Φάκελος \ κύριο όνομα . επέκταση

Πχ c:\temp\myfiles\data10.txt

- Τα ονόματα των αρχείων ακολουθούν τους κανόνες του λειτουργικού συστήματος
- Όταν το μέσο και ο φάκελος είναι το **τρέχον** μέσο και ο τρέχον φάκελος μιας εφαρμογής τότε το αρχείο αναγνωρίζεται από την εφαρμογή μόνο με το όνομά του και την επέκταση.
- Η επέκταση είναι προαιρετική.

Σημείωση. Στην γλώσσα C ο χαρακτήρας ` \ ` χρησιμοποιείται για την εισαγωγή ειδικών χαρακτήρων. Για να περάσουμε το όνομα ενός αρχείου σε ένα αλφαριθμητικό χρησιμοποιούμε το ` \ \ ` , π.χ.

```
char *filename="c:\\temp\\myfiles\\data10.txt"
```

Τα αρχεία αποθηκεύουν την πληροφορία σε δυαδική μορφή (Bits) . Αν οι πληροφορίες έχουν αποθηκευτεί έτσι ώστε το κάθε byte να αντιστοιχεί σε ένα χαρακτήρα ASCII, τότε τα αρχεία αυτά ονομάζονται **αρχεία κειμένου** (text files)

Αρχεία κειμένου στη C

Για την σύνδεση ενός αρχείου με το πρόγραμμά μας ορίζουμε μια ροή (stream) με τον τύπο δεδομένων FILE (**stdio.h**). Το πρόγραμμά μας θα προσπελαύνει ένα αρχείο μέσω ενός δείκτη σε δεδομένα του τύπου FILE

```
FILE *fp;
```

Ο δείκτης αρχείου παίρνει τιμή (δηλαδή συνδέεται με ένα συγκεκριμένο αρχείο στο μέσο αποθήκευσης) με την συνάρτηση fopen()

```
fp = fopen(char *filename, char *mode);
```

Filename : όνομα αρχείου στο μέσο αποθήκευσης

mode : κατάσταση ροής αρχείου

"r" : άνοιγμα για διάβασμα (read)

"w" : δημιουργία και άνοιγμα για γράψιμο (write) – σβήνει προϋπάρχον αρχείο

"a" : άνοιγμα για συμπλήρωση δεδομένων στο τέλος του αρχείου (append)

Επιστρέφει στο δείκτη τη τιμή NULL (0) αν δεν μπορεί να ανοίξει το αρχείο

Μπορούν να χρησιμοποιηθούν και οι καταστάσεις "rt", "wt", "at" (text files) για το διαχωρισμό τους από τα δυαδικά αρχεία (binary files) χαρακτήρων που δηλώνονται με την κατάσταση "rb", "wb", "ab". Στα περισσότερα συστήματα δεν υπάρχει διαφορά μεταξύ των καταστάσεων t και b.

Εγγραφή δεδομένων σε αρχείο

```
main()  
{
```

Εγγραφή Χαρακτήρα

```
putc(char , FILE *);  
fputc(char, FILE *);
```

Εγγραφή αλφαριθμητικού

```
fputs(char *, FILE *)
```

```
FILE *fout;  
char *s="Αυτό το κείμενο\n θα γραφεί στο αρχείο\n";  
int i,n;  
  
fout=fopen("testfile.txt","w");  
  
fputc('H',fout); fputc('i',fout); fputc('\n',fout);  
fputs(s,fout);  
  
fclose(fout);
```

Άνοιγμα Αρχείου
για γράψιμο

Εγγραφή δεδομένων

Κλείσιμο Αρχείου

Ένα αρχείο πρέπει να κλείνει με το τέλος της εγγραφής.

fclose(FILE *)

- Στέλνονται στο αρχείο και τα τελευταία δεδομένα που είναι ακόμα στο buffer της ροής
- Εγγράφεται το σύμβολο EOF (-1) που δηλώνει το τέλος του αρχείου
- Διακόπτεται η σύνδεση του δείκτη με το αρχείο.

fflush(FILE *) : στέλνει τα δεδομένα που είναι ακόμα στο buffer της ροής στο αρχείο

Βλ. κώδικα prog57witetext.c

Εγγραφή δεδομένων - fprintf

Εγγραφή Μορφοποιημένου αλφαριθμητικού

```
fprintf (FILE *, . . . . . );
```

Λειτουργεί όπως η `printf` με έξοδο το αρχείο `FILE` αντί της κονσόλας.

**Βλ. κώδικα
prog58writedata.c**

Συμπλήρωση στοιχείων στο τέλος του αρχείου (append)

- Άνοιγμα αρχείου σε κατάσταση “a”
- Εγγραφή νέων δεδομένων
- Κλείσιμο αρχείου
- **Αν το αρχείο δεν υπάρχει τότε δημιουργείται!**

**Βλ. κώδικα
prog58append.c**

Διάβασμα χαρακτήρων/αλφαριθμητικών από αρχείο

Διάβασμα Χαρακτήρα

`char c=getc(FILE *);` ή `char c=fgetc(FILE *);`

Διάβασμα αλφαριθμητικού

`char * fgets(char *s ,int n, FILE *)`: Διαβάζει από το αρχείο το αλφαριθμητικό `s`. Διαβάζονται το πολύ `n` χαρακτήρες. Επιστρέφεται ο δείκτης του αλφαριθμητικού `s` ή η τιμή `NULL` σε περίπτωση λάθους.

```
/*prog60readtext.c*/  
#include <stdio.h>  
#include <string.h>
```

```
main()  
{  
    FILE *fout;  
    char c;  
    fout=fopen("testfile.txt","r");  
    while(!feof(fout)) {c=fgetc(fout); putc(c,stdout);}  
    fclose(fout);  
    getchar();  
}
```

Διάβασμα αρχείου
χαρακτήρα-χαρακτήρα

```
main()  
{  
    FILE *fout;  
    char str[50];  
    fout=fopen("testfile.txt","r");  
    while(!feof(fout)) {  
        fgets(str,49,fout);  
        if(str==NULL) break; else puts(str);  
    }  
    fclose(fout);  
    getchar();  
}
```

Διάβασμα αρχείου με
αλφαριθμητικό 50
χαρακτήρων

Bool feof(FILE *)

Ελέγχει αν έφτασε το τέλος του αρχείου

Βλ. κώδικα prog57readtext.c

Διάβασμα δεδομένων με την fscanf

Διάβασμα Μορφοποιημένου αλφαριθμητικού : `int fscanf(FILE *, . . .)`;

Λειτουργεί όπως η `scanf` με είσοδο το αρχείο FILE αντί του πληκτρολογίου (stdin).

Επιστρέφει τον αριθμό των δεδομένων που διαβάστηκαν σωστά.

```
main()
{
    double celsiou, fahrenheit;
    FILE *f;

    f=fopen("c:\\temp\\temperatures.dat", "w");
    for(celsiou=-10; celsiou<41; celsiou++){
        fahrenheit=9*celsiou/5+32;
        fprintf(f, "%3.1f   %3.1f\n", celsiou, fahrenheit);
    }
    fclose(f);
}
```

Εγγραφή
δεδομένων

Διάβασμα
δεδομένων

```
main()
{
    double celsiou, fahrenheit;
    FILE *f;
    int n;

    f=fopen("c:\\temp\\temperatures.dat", "r");
    while(!feof(f)){
        n=fscanf(f, "%1f %1f\n", &celsiou, &fahrenheit);
        if(n!=2) {printf("File Error"); break;}
        printf("%3.1f   %3.1f\n", celsiou, fahrenheit);
    }
    fclose(f);
    getchar();
}
```

Βλ. κώδικα `prog59fprintf.c`
`prog59fscanf.c`

Ασκήσεις

Άσκηση 5.60. Γράψτε σε ένα αρχείο τα δεδομένα (x,y) όπου $y=\sin(x)*\cos(x+\pi/3)$ και το x να παίρνει τιμές στο διάστημα $[0,20]$ με βήμα $\Delta x=0.1$. Στη συνέχεια χρησιμοποιείστε ένα πρόγραμμα σχεδίασης δεδομένων (πχ το origin) και σχεδιάστε τα δεδομένα στο επίπεδο $x-y$.

Άσκηση 5.61. Γράψτε ένα πρόγραμμα που να δημιουργεί ένα αρχείο 100 γραμμών με τυχαία δεδομένα x,y,z (τρεις στήλες). Στη συνέχεια γράψτε ένα αρχείο που να διαβάζει το προηγούμενο και να υπολογίζει τους μέσους όρους για τα δεδομένα x, y και z .

Άσκηση 5.70. Γράψτε με τον κειμενογράφο “notepad” των windows το διπλανό κείμενο και αποθηκεύστε το με ένα όνομα.
Κάντε ένα πρόγραμμα που να διαβάζει το παραπάνω αρχείο, να βάζει τα δεδομένα σε κατάλληλες μεταβλητές (string, int, float) και να τα εκτυπώνει στην οθόνη.

Onoma	AEM	Vathmos
Giannis	1123	8.75
Maria	1234	6.20
Nikos	1456	9.70
Eleni	1234	4.50
Katerina	1238	9.95
Vasilis	1345	2.75

(σημείωση: κάντε αρχικά το πρόγραμμα χωρίς να συμπεριλάβετε την επικεφαλίδα “onoma AEM vathmos”)

Άσκηση 5.71. Γράψτε με τον κειμενογράφο “notepad” των windows ένα κείμενο και αποθηκεύστε το με ένα όνομα.
Κάντε ένα πρόγραμμα που να διαβάζει το παραπάνω αρχείο (χαρακτήρα προς χαρακτήρα) και να το εκτυπώνει στην οθόνη. Επίσης, ταυτόχρονα, δημιουργήστε ένα δεύτερο αρχείο που να περιέχει το ίδιο κείμενο αλλά με όλα τα γράμματα κεφαλαία.