



Δ.Π.Θ Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Αρχεία

Dr. Αθανάσιος Μπαλαφούτης
Εργαστηριακό Διδακτικό Προσωπικό
Τομέας Συστημάτων Παραγωγής
Εργαστήριο Ρομποτικής και Αυτοματισμών
abalafou@pme.duth.gr
Γραφείο 304, τηλ.: 25410 – 79892

Τι είναι ένα Αρχείο;



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Μια **συλλογή δεδομένων με συγκεκριμένο όνομα**, που συνήθως αποθηκεύεται σε δευτερεύουσες μονάδες αποθήκευσης (μη πτητικές), όπως σκληροί δίσκοι (HDD), δίσκοι SSD, κάρτες μνήμης, USB flash drives, ROM κ.ά..

Παραδείγματα:

- Δεδομένα με στοιχεία υπαλλήλων μιας εταιρείας
- Έγγραφο κειμένου στο Microsoft Word
- Ταινία βίντεο
- Αρχείο ήχου ή μουσικής

Δομή και χαρακτηριστικά αρχείου

- Ένα αρχείο είναι μια **συλλογή από δεδομένα**, αποθηκευμένα σε **μορφή ροής από bytes**.
- Τα bytes είναι **λογικά συνεχόμενα** (μπορεί να μην είναι φυσικά συνεχόμενα στον δίσκο).
- Προσπέλαση στα δεδομένα: **σειριακή και τυχαία** με τη βοήθεια ενός δείκτη.
- Κάθε αρχείο αποτελείται από 3 μέρη:
 - Την Αρχή, όπου αποθηκεύονται τα Meta-data (πληροφορίες σχετικά με το αρχείο)
 - Τα πραγματικά δεδομένα (ακολουθία από bytes)
 - Το Τέλος του αρχείου



Σε κάθε βήμα της ανάγνωσης του αρχείου ελέγχουμε αν έχουμε φτάσει στο τέλος του EOF



- **Αρχεία Κειμένου** (text files) - περιέχουν μόνο χαρακτήρες ASCII
 - Πηγαίος κώδικας (π.χ.: προγράμματα γραμμένα σε C)
 - Αρχεία κειμένου .txt
 - xml, json, csv
- **Δυαδικά αρχεία** (binary files) - δεν περιέχει απλό αναγνώσιμο κείμενο, αλλά δεδομένα σε δυαδική μορφή (0 και 1), τα οποία διαβάζονται από προγράμματα και όχι από τον άνθρωπο απευθείας.
 - Εικόνες
 - Video
 - Αρχεία ήχου
 - Εκτελέσιμα αρχεία



- **Αρχεία Κειμένου** (text files) - περιέχουν μόνο χαρακτήρες ASCII
 - Πηγαίος κώδικας (π.χ.: προγράμματα γραμμένα σε C)
 - Αρχεία κειμένου .txt
 - xml, json, csv
- **Δυαδικά αρχεία** (binary files) - δεν περιέχει απλό αναγνώσιμο κείμενο, αλλά δεδομένα σε δυαδική μορφή (0 και 1), τα οποία διαβάζονται από προγράμματα και όχι από τον άνθρωπο απευθείας.
 - Εικόνες
 - Video
 - Αρχεία ήχου
 - Εκτελέσιμα αρχεία

Σε ποιον τύπο ανήκει ένα αρχείο .docx του MS-Word?

Βασικές λειτουργίες στα Αρχεία



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

- **Άνοιγμα** (Open): Άνοιγμα του αρχείου για ανάγνωση ή εγγραφή
- **Ανάγνωση** (Read): Ανάγνωση του περιεχομένου (ολόκληρου ή κατά γραμμή/κομμάτι)
- **Εγγραφή** (Write): Εγγραφή (ή αντικατάσταση) περιεχομένου
- **Κλείσιμο** (Close): Κλείσιμο του αρχείου μετά την επεξεργασία

Άνοιγμα και Κλείσιμο Αρχείου



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Ο χειρισμός των αρχείων γίνεται με την βοήθεια ενός δείκτη τύπου **FILE** που είναι ορισμένος στη βιβλιοθήκη **<stdio.h>**

```
FILE *fptr; // fptr δείκτης σε αρχείο
```

Άνοιγμα και Κλείσιμο Αρχείου



Ο χειρισμός των αρχείων γίνεται με την βοήθεια ενός δείκτη τύπου **FILE** που είναι ορισμένος στη βιβλιοθήκη **<stdio.h>**

```
FILE *fptr; // fptr δείκτης σε αρχείο
```

Το άνοιγμα ενός αρχείου γίνεται με τη βοήθεια της συνάρτησης **fopen()**:

```
fptr = fopen("example.txt", "r"); // Άνοιγμα του αρχείου σε λειτουργία ανάγνωσης ("r")
```

Άνοιγμα και Κλείσιμο Αρχείου



Ο χειρισμός των αρχείων γίνεται με την βοήθεια ενός δείκτη τύπου **FILE** που είναι ορισμένος στη βιβλιοθήκη **<stdio.h>**

```
FILE *fptr; // fptr δείκτης σε αρχείο
```

Το άνοιγμα ενός αρχείου γίνεται με τη βοήθεια της συνάρτησης **fopen()**:

```
fptr = fopen("example.txt", "r"); // Άνοιγμα του αρχείου σε λειτουργία ανάγνωσης ("r")
```

Το κλείσιμο ενός αρχείου γίνεται με τη βοήθεια της συνάρτησης **fclose()**:

```
fclose(fptr); // Κλείσιμο του αρχείου
```

Ορίσματα συνάρτησης fopen()



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Το δεύτερο όρισμα της συνάρτησης fopen() υποδηλώνει τον τύπο λειτουργίας (για ποιο λόγο ανοίγουμε το αρχείο)

```
fptr = fopen("example.txt", "r"); // Άνοιγμα του αρχείου σε λειτουργία ανάγνωσης ("r")
```

Υπάρχουν οι παρακάτω διαφορετικοί τύποι (modes):

- **"r"**: Άνοιγμα του αρχείου για ανάγνωση
- **"w"**: Άνοιγμα αρχείου για εγγραφή (αν το αρχείο υπάρχει ήδη, διαγράφει το περιεχόμενό του)
- **"a"**: Άνοιγμα για εγγραφή νέου περιεχομένου στο τέλος του αρχείου (δεν διαγράφει το υπάρχον)
- **"rb"**: Άνοιγμα δυαδικού αρχείου για ανάγνωση
- **"wb"**: Άνοιγμα δυαδικού αρχείου για εγγραφή (αν το αρχείο υπάρχει ήδη, διαγράφει το περιεχόμενό του)

Σε περίπτωση που αποτύχει το άνοιγμα ενός αρχείου ο δείκτης επιστρέφει τιμή **NULL**

Παράδειγμα χρήσης fopen() και fclose()



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>

int main() {
    FILE *file;

    // Άνοιγμα του αρχείου σε λειτουργία ανάγνωσης ("r")
    file = fopen("example.txt", "r");

    if (file == NULL) {
        printf("Σφάλμα: Δεν ήταν δυνατό να ανοίξει το αρχείο.\n");
        return 1;
    }

    // ... κώδικας για την επεξεργασία του αρχείου ...
    fclose(file);
    return 0;
}
```

Συναρτήσεις Ανάγνωσης Αρχείου



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

fgets(): Διαβάζει μια γραμμή του αρχείου

```
char line[100];  
fgets(line, sizeof(line), file); // Διαβάζει μέχρι 99 χαρακτήρες ή μέχρι '\n'
```

Συναρτήσεις Ανάγνωσης Αρχείου



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

fgets(): Διαβάζει μια γραμμή του αρχείου

```
char line[100];  
fgets(line, sizeof(line), file); // Διαβάζει μέχρι 99 χαρακτήρες ή μέχρι '\n'
```

fgetc(): Διαβάζει έναν χαρακτήρα

```
char ch;  
ch = fgetc(file);
```

Συναρτήσεις Ανάγνωσης Αρχείου



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

fgets(): Διαβάζει μια γραμμή του αρχείου

```
char line[100];  
fgets(line, sizeof(line), file); // Διαβάζει μέχρι 99 χαρακτήρες ή μέχρι '\n'
```

fgetc(): Διαβάζει έναν χαρακτήρα

```
char ch;  
ch = fgetc(file);
```

fscanf(): Μορφοποιημένη ανάγνωση

```
int age;  
char name[50];  
fscanf(file, "%s %d", name, &age); // Π.χ. αν το αρχείο έχει: "Nikos 25"
```

Παράδειγμα Ανάγνωσης Αρχείου



```
#include <stdio.h>
int main() {
    FILE *file;
    char line[100];
    file = fopen("input.txt", "r");
    if (file == NULL) {
        printf("Σφάλμα: Δεν βρέθηκε το αρχείο.\n");
        return 1;
    }
    // Ανάγνωση γραμμή-γραμμή και εμφάνιση στην οθόνη
    while (fgets(line, sizeof(line), file)) {
        printf("%s", line);
    }
    fclose(file);
    return 0;
}
```

Αρχείο input.txt

```
Νίκος 1
Μαρία 2
Φώτης 3
Άννα 4
```

Τι θα εκτυπώσει η printf() ?

Παράδειγμα Ανάγνωσης Αρχείου



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
int main() {
    FILE *file;
    char line[100];
    file = fopen("input.txt", "r");
    if (file == NULL) {
        printf("Σφάλμα: Δεν βρέθηκε το αρχείο.\n");
        return 1;
    }
    // Ανάγνωση γραμμή-γραμμή και εμφάνιση στην οθόνη
    while (fgets(line, sizeof(line), file)) {
        printf("%s", line);
    }
    fclose(file);
    return 0;
}
```

Αρχείο input.txt

```
Νίκος 1
Μαρία 2
Φώτης 3
Άννα 4
```

Τι θα εκτυπώσει η printf() ?

```
Νίκος 1
Μαρία 2
Φώτης 3
Άννα 4
```

Παράδειγμα Ανάγνωσης Αρχείου

```
#include <stdio.h>
int main() {
    FILE *file;
    char name[100];
    int num;
    file = fopen("input.txt", "r");
    if (file == NULL) {
        printf("Σφάλμα: Δεν βρέθηκε το αρχείο.\n");
        return 1;
    }
    while (fscanf(file, "%s %d", name, &num) != EOF) {
        printf("%d, %s \n", num, name);
    }
    fclose(file);
    return 0;
}
```

Αρχείο input.txt

```
Νίκος 1
Μαρία 2
Φώτης 3
Άννα 4
```

Τι θα εκτυπώσει η printf() ?

Παράδειγμα Ανάγνωσης Αρχείου

```
#include <stdio.h>
int main() {
    FILE *file;
    char name[100];
    int num;
    file = fopen("input.txt", "r");
    if (file == NULL) {
        printf("Σφάλμα: Δεν βρέθηκε το αρχείο.\n");
        return 1;
    }
    while (fscanf(file, "%s %d", name, &num) != EOF) {
        printf("%d. %s \n", num, name);
    }
    fclose(file);
    return 0;
}
```

Αρχείο input.txt

```
Νίκος 1
Μαρία 2
Φώτης 3
Άννα 4
```


Τι θα εκτυπώσει η printf() ?

1. Νίκος
2. Μαρία
3. Φώτης
4. Άννα

Χρήση συνάρτησης feof()



```
#include <stdio.h>
int main() {
    FILE *file;
    char name[100];
    int num;
    file = fopen("input.txt", "r");
    if (file == NULL) {
        printf("Σφάλμα: Δεν βρέθηκε το αρχείο.\n");
        return 1;
    }
    while (fscanf(file, "%s %d", name, &num) != EOF) {
        printf("%d. %s \n", num, name);
    }
    fclose(file);
    return 0;
}
```



```
while (!feof(file)) {
    fscanf(file, "%s %d", name, &num)
    printf("%d. %s \n", num, name);
}
```

Συναρτήσεις Εγγραφής σε Αρχείο



fputs(): Εγγράφει μια ολόκληρη συμβολοσειρά (string)

```
fputs("Καλημέρα κόσμε!\n", file);
```

Συναρτήσεις Εγγραφής σε Αρχείο



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

fputs(): Εγγράφει μια ολόκληρη συμβολοσειρά (string)

```
fputs("Καλημέρα κόσμε!\n", file);
```

fputc(): Εγγραφή ενός μόνο χαρακτήρα

```
fputc('A', file);  
fputc('\n', file);
```

Συναρτήσεις Εγγραφής σε Αρχείο



fputs(): Εγγράφει μια ολόκληρη συμβολοσειρά (string)

```
fputs("Καλημέρα κόσμε!\n", file);
```

fputc(): Εγγραφή ενός μόνο χαρακτήρα

```
fputc('A', file);  
fputc('\n', file);
```

fprintf(): Εγγραφή μορφοποιημένου κειμένου

```
int age = 25;  
fprintf(file, "Η ηλικία είναι: %d\n", age);
```

Παράδειγμα Εγγραφής Αρχείου



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
#include <stdio.h>
int main() {
    FILE *file = fopen("output.txt", "w");

    if (file == NULL) {
        printf("Σφάλμα στο άνοιγμα αρχείου!\n");
        return 1;
    }
    fprintf(file, "Γεια σου, κόσμε!\n");
    fputs("Αυτή είναι μια δεύτερη γραμμή.\n", file);
    fputc('A', file);
    fputc('\n', file);

    fclose(file);
    return 0;
}
```

Παράδειγμα Εγγραφής Αρχείου

```
#include <stdio.h>
int main() {
    FILE *file = fopen("output.txt", "w");

    if (file == NULL) {
        printf("Σφάλμα στο άνοιγμα αρχείου!\n");
        return 1;
    }
    fprintf(file, "Γεια σου, κόσμε!\n");
    fputs("Αυτή είναι μια δεύτερη γραμμή.\n", file);
    fputc('A', file);
    fputc('\n', file);

    fclose(file);
    return 0;
}
```

Αρχείο output.txt

Γεια σου, κόσμε!
Αυτή είναι μια δεύτερη γραμμή.
A

Άσκηση 9.1



Να αποθηκεύσετε σε ένα κειμένου (text file) τους παρακάτω αριθμούς:

5 96 87 78 93 21 4 92 82 85 87 6 72 69 85 75 81 73

Οι αριθμοί είναι γραμμένοι κατά ομάδες. Ο 1ος αριθμός κάθε ομάδας δείχνει το πλήθος των μελών της ομάδας που ακολουθεί αμέσως μετά τον αριθμό αυτό.

Στη συνέχεια να γραφεί πρόγραμμα σε γλώσσα C που θα βρίσκει και θα εμφανίζει τον μέσο όρο κάθε ομάδας.

Άσκηση 9.1 - findMean



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

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

double findMean(int arr[], int n) {
    int sum = 0;
    for (int i = 0; i < n; i++) {
        sum += arr[i];
    }
    return (double)sum / n;
}
```

Άσκηση 9.1 - main (1/2)

Αρχείο 09_01.txt

```
int main() {  
  
    FILE *file = fopen("09_01.txt", "r");  
  
    if (file == NULL) {  
        printf("Σφάλμα: Δεν βρέθηκε το αρχείο. \n");  
        return 1;  
    }  
}
```

```
5 96 87 78 93 21  
4 92 82 85 87  
6 72 69 85 75 81 73
```

Άσκηση 9.1 - main (2/2)



```
int num;
while (fscanf(file, "%d", &num) != EOF) {
    int teamSize = num;
    int *teamMembers = (int *)malloc(teamSize * sizeof(int));

    for (int i = 0; i < teamSize; i++)
        fscanf(file, "%d", &teamMembers[i]);

    double mean = findMean(teamMembers, teamSize);
    printf("Μέση τιμή ομάδας: %.2f\n", mean);

    free(teamMembers);
}
fclose(file);
return 0;
}
```

Άσκηση 9.2

Να δημιουργήσετε ένα αρχείο κειμένου (text file) με όνομα `cars.txt` κάθε γραμμή του οποίου θα πρέπει να περιλαμβάνει τα εξής στοιχεία ενός αυτοκινήτου:

- Αριθμός αυτοκινήτου (αλφαριθμητικό 7 χαρακτήρων)
- Διανυθέντα χιλιόμετρα (θετικός ακέραιος τύπου `long int`)
- Καταναλωθέντα καύσιμα (θετικός αριθμός, τύπου `double`, σε λίτρα)

Στη συνέχεια να γραφεί πρόγραμμα σε γλώσσα C που:

1. Θα εμφανίζει τα περιεχόμενα του αρχείου στην οθόνη, μία γραμμή του αρχείου ανά γραμμή της οθόνης
2. Θα εμφανίζει στο τέλος το σύνολο χιλιομέτρων και καυσίμων, για όλο το αρχείο
3. Θα εμφανίζει το αυτοκίνητο με τη μεγαλύτερη κατανάλωση καυσίμου, με ακρίβεια 10^{-3} . Σε περίπτωση που υπάρχουν περισσότερα από ένα αυτοκίνητα με τη μέγιστη κατανάλωση καυσίμου τότε θα πρέπει να εμφανίζονται όλα.

Άσκηση 9.2 - CarRecord



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

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

typedef struct {
    char carNumber[8];
    long distance;
    double fuel;
} CarRecord;
```

```
int main() {

    FILE *file = fopen("files/cars.txt", "r");
    if (file == NULL) {
        printf("Σφάλμα: Δεν βρέθηκε το αρχείο.\n");
        return 1;
    }

    CarRecord car[100];
    int count = 0;
    long totalDistance = 0;
    double totalFuel = 0.0;
    double maxFuel = 0.0;
```

Άσκηση 9.2 - main (1/2)



```
while (fscanf(file, "%s %ld %lf", &car[count].carNumber, &car[count].distance,
    &car[count].fuel) != EOF) {
    totalDistance += car[count].distance;
    totalFuel += car[count].fuel;
    if (car[count].fuel > maxFuel) {
        maxFuel = car[count].fuel;
    }
    count++;
}
fclose(file);

for (int i = 0; i < count; i++)
    printf("%s\t%ld\t%.2f\n", car[i].carNumber, car[i].distance, car[i].fuel);
```

Άσκηση 9.2 - main (2/2)



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

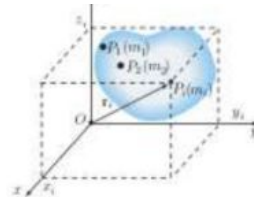
```
printf("Συνολική απόσταση: %ld Km\n", totalDistance);
printf("Συνολικά καύσιμα: %.2f λίτρα\n", totalFuel);

printf("Μεγαλύτερη κατανάλωση καυσίμων (%.2f λίτρα) για τα αυτοκίνητα:\n", maxFuel);
for (int i = 0; i < count; i++) {
    if (car[i].fuel == maxFuel) {
        printf("Αριθμός: %s, Απόσταση: %ld Km, Κατανάλωση: %.2f λίτρα\n",
            car[i].carNumber, car[i].distance, car[i].fuel);
    }
}
return 0;
}
```

Άσκηση 9.3



Να ορίσετε μία δομή με όνομα `mass_point` που θα χρησιμοποιηθεί για τη διαχείριση υλικών σημείων ενός στερεού σώματος. Τα μέλη της δομής είναι οι τρεις συντεταγμένες του υλικού σημείου στο χώρο και η μάζα του υλικού σημείου, όπως φαίνεται στο σχήμα (απεικονίζονται τα υλικά σημεία P_1, P_2, \dots, P_i με μάζες αντίστοιχα m_1, m_2, \dots, m_i). Όλες οι τιμές είναι τύπου `double`. Οι τύποι αντιστοιχούν στις ροπές αδρανείας $I_{xOy}, I_{yOz}, I_{zOx}$.



$$I_{xOy} = \sum_i m_i z_i^2, \quad I_{yOz} = \sum_i m_i x_i^2, \quad I_{zOx} = \sum_i m_i y_i^2$$

Ένα σειριακό αρχείο με όνομα `poly.txt` περιέχει άγνωστο πλήθος γραμμών που αντιστοιχούν στα υλικά σημεία ενός στερεού σώματος. Κάθε γραμμή του αρχείου περιέχει τέσσερις τιμές τύπου `double` που αντιστοιχούν στις τρεις συντεταγμένες (μέλη-δεδομένα) και στη μάζα ενός υλικού σημείου (μέλη-δεδομένα). Να οριστεί ένας δυναμικός πίνακας δομών τύπου `mass_point`, με πλήθος στοιχείων ίσο με το πλήθος των γραμμών του αρχείου. Τα δεδομένα στον πίνακα δομών θα εισάγονται μέσω προσπέλασης του σειριακού αρχείου `poly.txt`.

Άσκηση 9.3



Ζητούνται τα παρακάτω — συνάρτηση `main()`:

1. να γίνει η εισαγωγή των δεδομένων από το αρχείο `poly.txt` στον πίνακα δομών, ελέγχοντας την ύπαρξη του αρχείου. Αν το αρχείο περιλαμβάνει περισσότερα από `N` υλικά σημεία να βρείτε και να εμφανίσετε το πλήθος των γραμμών του αρχείου που δεν θα συμπεριληφθούν στον πίνακα δομών. Αν το αρχείο περιλαμβάνει λιγότερες από `N` γραμμές να βρείτε και να εμφανίσετε το πλήθος των γραμμών του αρχείου που θα περιέχει ο πίνακας δομών.
2. Στη συνέχεια να βρεθούν και να εμφανιστούν οι ροπές αδρανείας I_{x0y} , I_{y0z} , I_{z0x} λαμβάνοντας υπόψη όλα τα στοιχεία του πίνακα δομών.

Άσκηση 9.3 - main (1/3)



```
#include <stdio.h>
#include <stdlib.h>
#define N 50

typedef struct {
    double x;
    double y;
    double z;
    double m;
} mass_point;
```

```
int main() {

    mass_point *points;
    int count = 0;
    int skipped = 0;
    double x, y, z, m;
    double Ix0y = 0.0, Iy0z = 0.0, Iz0x = 0.0;

    FILE *file = fopen("files/poly.txt", "r");
    if (file == NULL) {
        printf("Σφάλμα: Δεν βρέθηκε το αρχείο.\n");
        return 1;
    }
}
```

Άσκηση 9.3 - main (2/3)



```
points = (mass_point *)malloc(N * sizeof(mass_point));
if (points == NULL) {
    printf("Σφάλμα: Αποτυχία δέσμευσης μνήμης\n");
    fclose(file);
    return 1;
}
while (fscanf(file, "%lf %lf %lf %lf", &x, &y, &z, &m) != EOF) {
    if (count < N) {
        points[count].x = x;
        points[count].y = y;
        points[count].z = z;
        points[count].m = m;
        count++;
    } else
        skipped++;
}
fclose(file);
```

Άσκηση 9.3 - main (3/3)



```
if (skipped > 0)
    printf("Το αρχείο είχε %d γραμμές που δεν χωρούσαν στον πίνακα.\n", skipped);
else
    printf("Ο πίνακας περιέχει %d γραμμές.\n", count);

for (int i = 0; i < count; i++) {
    IxOy += points[i].m * (points[i].z * points[i].z);
    IyOz += points[i].m * (points[i].x * points[i].x);
    IzOx += points[i].m * (points[i].y * points[i].y);
}
printf("Ροπές αδρανείας:\n");
printf("IxOy = %.3lf\n", IxOy);
printf("IyOz = %.3lf\n", IyOz);
printf("IzOx = %.3lf\n", IzOx);
free(points);
return 0;
}
```