

# Άσκηση 9.4



Δίνεται αρχείο κειμένου (text file), με όνομα `production.txt`, κάθε γραμμή του οποίου περιλαμβάνει:

- Έναν ακέραιο θετικό αριθμό `[1-100]` που αντιστοιχεί στον κωδικό ενός μεταλλικού εξαρτήματος που υφίσταται κατεργασία σε κάποια εργαλειομηχανή μιας μονάδας παραγωγής μεταλλικών εξαρτημάτων
- Έναν ακέραιο θετικό αριθμό που αντιστοιχεί στον κωδικό της εργαλειομηχανής `[1-10]`
- Μια ακολουθία `10` ακεραίων θετικών αριθμών (περιοχή τιμών `[1, 100]` ) καθένας από τους οποίους αντιστοιχεί στην ποσότητα τεμαχίων του μεταλλικού εξαρτήματος που κατεργάστηκε η συγκεκριμένη εργαλειομηχανή σε `10` χρονικές περιόδους λειτουργίας.

Μία γραμμή του αρχείου είναι η παρακάτω (αντιστοιχεί στο μεταλλικό εξάρτημα με κωδικό `17` και στην εργαλειομηχανή με κωδικό `9`) :

`17 2 97 1 16 36 18 77 4 33 86 35`

Να γραφεί μια δομή (struct) που θα περιλαμβάνει ως μέλη τα στοιχεία που περιέχει μια οποιαδήποτε γραμμή του αρχείου καθώς και ένα επιπλέον μέλος που αντιστοιχεί στο συνολικό αριθμό των τεμαχίων (δηλαδή των `10` τελευταίων αριθμών).

Ένας πίνακας δομών του παραπάνω τύπου μεγέθους `N` ( `N` = γνωστό) γεμίζει με δεδομένα από το αρχείο.

# Άσκηση 9.4

Να γραφεί πρόγραμμα σε γλώσσα C το οποίο:

1. να καταχωρεί τα δεδομένα στον πίνακα διαβάζοντας το αρχείο (απαιτείται έλεγχος για την ύπαρξη του αρχείου). Απαιτείται έλεγχος αν το πλήθος των εγγραφών από το αρχείο υπερβαίνει ή όχι το μέγεθος  $N$ .
2. χρησιμοποιώντας όλα τα στοιχεία του πίνακα δομών θα βρίσκει και θα εμφανίζει το συνολικό πλήθος τεμαχίων που έχει κατεργαστεί η κάθε μία από τις εργαλειομηχανές της μονάδας.
3. χρησιμοποιώντας όλα τα στοιχεία του πίνακα δομών θα βρίσκει και θα εμφανίζει το συνολικό πλήθος τεμαχίων για κάθε κωδικό μεταλλικού εξαρτήματος που υπάρχουν στο αρχείο.

# Άσκηση 9.4 - main (1/4)



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

#define N 1000

typedef struct {
    int part_code;
    int machine_code;
    int quantities[10];
    int total_quantity;
} production_record;
```

```
int main() {
    FILE *file;
    production_record records[N];
    int part, machine, quantities[10];
    int count = 0;
    int skipped = 0;

    int machine_totals[11] = {0};
    int part_totals[101] = {0};

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

## Άσκηση 9.4 - main (2/4)



```
while (fscanf(file, "%d %d", &part, &machine) != EOF) {  
  
    for (int i = 0; i < 10; i++)  
        fscanf(file, "%d", &quantities[i]);  
    if (count < N) {  
        records[count].part_code = part;  
        records[count].machine_code = machine;  
        records[count].total_quantity = 0;  
        for (int i = 0; i < 10; i++) {  
            records[count].quantities[i] = quantities[i];  
            records[count].total_quantity += quantities[i];  
        }  
        count++;  
    } else  
        skipped++;  
}  
fclose(file);
```

## Άσκηση 9.4 - main (3/4)



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
if (skipped > 0)
    printf("Το αρχείο είχε %d επιπλέον εγγραφές που αγνοήθηκαν.\n", skipped);
else
    printf("Διαβάστηκαν %d εγγραφές.\n", count);

for (int i = 0; i < count; i++) {
    int machine_id = records[i].machine_code;
    int part_id = records[i].part_code;
    machine_totals[machine_id] += records[i].total_quantity;
    part_totals[part_id] += records[i].total_quantity;
}
```

## Άσκηση 9.4 - main (4/4)



Δ.Π.Θ

Δομημένος Προγραμματισμός

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

```
printf("Συνολικό πλήθος τεμαχίων ανά εργαλειομηχανή:\n");
for (int i = 1; i <= 10; i++)
    printf("Εργαλειομηχανή %d: %d τεμάχια\n", i, machine_totals[i]);

printf("Συνολικό πλήθος τεμαχίων ανά μεταλλικό εξάρτημα:\n");
for (int i = 1; i < 100; i++)
    printf("Κωδικός εξαρτήματος %d: %d τεμάχια\n", i, part_totals[i]);
return 0;
}
```