

Άσκηση 1

Να ορίσετε τη δομή employee με πεδία: μικρό όνομα, επίθετο, ηλικία και μισθός. Να γραφεί ένα πρόγραμμα το οποίο να χρησιμοποιεί τη δομή employee για να διαβάσει τα στοιχεία 50 υπαλλήλων και να τα αποθηκεύσει σε έναν πίνακα. Το πρόγραμμα να εμφανίζει το επίθετο του υπαλλήλου με τον υψηλότερο μισθό και του υπαλλήλου με τη μεγαλύτερη ηλικία. (Σημ. Αν παραπάνω από ένας υπάλληλοι έχουν τον ίδιο υψηλότερο μισθό ή μεγαλύτερη ηλικία, το πρόγραμμα να εμφανίζει τα δεδομένα του πρώτου από αυτούς).

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

#define SIZE 50

struct employee
{
    char fname[50]; /* με χρήση ένθετης δομής ονοματεπώνυμο? */
    char lname[50]; /* ή με χρήση απλού δείκτη αντί πίνακα? */
    int age;
    int salary;
};

int main(void)
{
    int i, max_age, max_sal, age_pos, sal_pos;
    struct employee empl[SIZE];
    max_age = max_sal = 0;
```

```
for(i=0; i<SIZE; i++)
{
    printf("\nFirst name: ");
    gets(empl[i].fname);

    printf("Last name: ");
    gets(empl[i].lname);

    printf("Age: ");
    scanf("%d", &empl[i].age);

    printf("Salary: ");
    scanf("%d", &empl[i].salary);

    if(empl[i].age > max_age)
    {
        max_age = empl[i].age;
        age_pos = i;
    }
    if(empl[i].salary > max_sal)
    {
        max_sal = empl[i].salary;
        sal_pos = i;
    }
    getchar();
}
```

```

printf("\n%s.%s is the oldest (%d y.o.)\n",empl[age_pos].fname,
      empl[age_pos].lname, max_age);
printf("%s.%s has the highest salary (%d€)\n",empl[sal_pos].fname,
      empl[sal_pos].lname, max_sal);
return 0;
}

```

### Άσκηση 1β

Να ξαναγράψετε το προηγούμενο πρόγραμμα με τις εξής αλλαγές: α) τα δεδομένα των φοιτητών θα είναι καταχωρισμένα στο αρχείο κειμένου "employees.dat" όπου το πλήθος των υπαλλήλων θα είναι στην πρώτη γραμμή και η κάθε επόμενη γραμμή θα περιλαμβάνει το ονοματεπώνυμο, την ηλικία και τον μισθό του κάθε υπαλλήλου, και β) το όνομα του αρχείου όπως και το είδος του στατιστικού που επιθυμούμε να εξαγάγουμε (μεγαλύτερο μισθό ή μεγαλύτερη ηλικία) θα τα δίνουμε από τη γραμμή εντολών.

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct employee
{
    char fname[50];
    char lname[50];
    int age;
    int salary;
};
int main(int argc, char *argv[])
{
    int i, N, choice, max_age = max_sal = 0, age_pos, sal_pos;
    struct employee *empl;
    FILE *fp;

    if(argc != 3)
    {
        printf("progr1 employees.dat output_choice\n");
        printf("where output_choice = 1 (2) for max age (max salary)\n");
        exit(-1);
    }
}

```

```

choice = atoi(argv[2]);
if((fp = fopen(argv[1], "r")) == NULL)
{
    fprintf(stderr, "Can't open file %s for reading!\n", argv[1]);
    exit(-1);
}
fscanf(fp, "%d\n", &N);
empl = (struct employee *) malloc(N*sizeof(struct employee));

for(i=0; i<N; i++)
{
    fscanf("%s%s%d\n", empl[i].fname, empl[i].lname, &empl[i].age,
          &empl[i].salary);

    if(empl[i].age > max_age)
    {
        max_age = empl[i].age;
        age_pos = i;
    }
    if(empl[i].salary > max_sal)
    {
        max_sal = empl[i].salary;
        sal_pos = i;
    }
}

```

```

if(choice == 1)
    printf("\n%s.%s is the oldest (%d y.o.)\n", empl[age_pos].fname,
          empl[age_pos].lname, max_age);
else
    printf("%s.%s has the highest salary (%d€)\n", empl[sal_pos].fname,
          empl[sal_pos].lname, max_sal);
return 0;
}

```

### Άσκηση 2

Έστω ότι σε μια εφαρμογή βιβλιοπωλείου θέλουμε να καταχωρίσουμε τα βιβλία σε έναν πίνακα δομών με πεδία: α) τον αριθμό ISBN του βιβλίου (συμβολοσειρά) που, λόγω της μοναδικότητάς του, θα αποτελεί και το «κλειδί» σε εφαρμογές αναζήτησης, β) το πλήθος του στην αποθήκη (δηλαδή, το stock), γ) τους συγγραφείς (επώνυμο 1ου, αρχικό γράμμα ονόματος 1ου, επώνυμο 2ου, αρχικό γράμμα ονόματος 2ου, ... κ.λπ.) , δ) τον τίτλο του βιβλίου, και ε) την χρονολογία έκδοσης. Να θεωρήσετε ότι ο πίνακας δομών δημιουργείται στην main (π.χ. με διάβασμα από το πληκτρολόγιο ή από αρχείο). Στη συνέχεια, να ταξινομήσετε τον πίνακα κατ'αλφαβητική σειρά ως προς το πεδίο των συγγραφέων και να τον αποθηκεύσει στο ASCII αρχείο books.dat.

Κατά τη δημιουργία του πίνακα να ζητείται το συνολικό πλήθος ή ο μέγιστος αριθμός βιβλίων προς εισαγωγή και να καλείται κατάλληλη συνάρτηση (π.χ. book\_insert) με δύο ορίσματα: τον πίνακα δομών και τη θέση εισαγωγής στον πίνακα. Η book\_insert θα προσθέτει το βιβλίο στον πίνακα αν δεν υπάρχει ήδη ή θα αυξάνει το πεδίο του stock κατά ένα και θα επιστρέφει (με return) την τιμή 1 αν έγινε νέα εισαγωγή ή την τιμή 0 αν υπήρχε το βιβλίο ώστε να μην αυξήσει η main τον μετρητή βιβλίων.

Επίσης να γραφεί και η συνάρτηση delete\_book για τη διαγραφή ενός βιβλίου από τον πίνακα αν το πεδίο stock = 1 είτε για να μειώσει το πεδίο stock κατά ένα.

Τέλος, να γραφεί και η συνάρτηση αναζήτησης retrieve με όρισμα έναν ISBN. Αν υπάρχει αυτό το βιβλίο να εμφανίζει όλες τις πληροφορίες σχετικά με το βιβλίο αυτό ειδικά να εμφανίζει μήνυμα ότι δεν υπάρχει.

### Άσκηση 2β

Να επαναλάβετε την προηγούμενη άσκηση με συνδεδεμένη λίστα αντί πίνακα δομών.