

## Εργαστήριο 9: Δομές και Αυτοαναφορικές Δομές

Σε αυτό το εργαστήριο θα μελετήσουμε τις δυνατότητες που μας προσφέρει η C για να ομαδοποιούμε δεδομένα χρησιμοποιώντας δομές. Μέσω των δομών θα κατασκευάσουμε συνδεδεμένες λίστες και δυαδικά δέντρα, τα οποία ονομάζονται αυτοαναφορικές δομές.

### Άσκηση 1: Δομές και συναρτήσεις - (point.c)

1.1 Κατασκευάστε το αρχείο point.c και ορίστε σε αυτό τη δομή point που αποθηκεύει τις συντεταγμένες (τύπου double) ενός σημείου στο διδιάστατο χώρο.

1.2 Ορίστε τη συνάρτηση:

```
struct point middle(struct point a, struct point b)
```

Η συνάρτηση θα υπολογίζει και θα επιστρέφει το σημείο που βρίσκεται στο μέσο του ευθυγράμμου τμήματος με άκρα τα σημεία a και b.

1.3 Υπολογίστε και τυπώστε το μέσο του ευθυγράμμου τμήματος με άκρα τα σημεία (1.2, 5.4) και (7.3, 1.8).

### Άσκηση 2: Δομές και δείκτες - (person.c)

2.1 Δημιουργήστε το αρχείο person.c και ορίστε τη δομή person που αποθηκεύει το όνομα, το επώνυμο και το πατρώνυμο ενός ατόμου.

```
struct person {  
    char *fname;  
    char *lname;  
    char *mname;  
};
```

2.2 Κατασκευάστε τη συνάρτηση:

```
struct person *person_init(char *firstname, char *lastname, char *middlename);
```

Η συνάρτηση θα δεσμεύει χώρο για μία δομή τύπου person, θα την αρχικοποιεί και θα επιστρέφει τη διεύθυνσή της. Καλέστε τη συνάρτηση από την main για να καταχωρίσετε τα στοιχεία του πατέρα σας.

2.3 Ορίστε τη συνάρτηση:

```
struct person *childof(struct person father, char *newname);
```

Η συνάρτηση θα καταχωρεί τα στοιχεία ενός παιδιού με μικρό όνομα newname, χρησιμοποιώντας τον πατέρα του father. Καλέστε τη συνάρτηση από την main για να καταχωρίσετε τα στοιχεία σας.

### Άσκηση 3: Συνδεδεμένες λίστες - (grades.c)

3.1 Κατασκευάστε το πρόγραμμα grades.c και ορίστε μία αυτοαναφορική δομή λίστας ακεραίων αριθμών.

```
typedef struct listnode *Listptr;  
  
struct listnode {  
    int data;  
    Listptr next;  
};
```

3.2 Κατασκευάστε τη συνάρτηση:

```
void insert_at_start(Listptr *ptr, int grade);
```

Η συνάρτηση θα προσθέτει έναν βαθμό στην αρχή της λίστας. Τροποποιήστε τη main για να διαβάξει βαθμούς από το πληκτρολόγιο και να τους προσθέτει στη λίστα.

3.3 Κατασκευάστε τη συνάρτηση:

```
float average(Listptr ptr);
```

Η συνάρτηση θα διασχίζει τα περιεχόμενα της λίστας και θα υπολογίζει τον μέσο όρο των βαθμών που είναι αποθηκευμένοι σε αυτήν.

### Άσκηση 4: Δυαδικά δένδρα - (tree.c)

4.1 Δημιουργήστε το αρχείο tree.c και ορίστε μία αυτοαναφορική δομή δυαδικού δένδρου ακεραίων αριθμών.

```
typedef struct tnode *Treeptr;  
  
struct tnode {  
    int data;  
    Treeptr left;  
    Treeptr right;  
};
```

4.2 Ένα ταξινομημένο δυαδικό δένδρο είναι ένα δυαδικό δένδρο στο οποίο κάθε κόμβος έχει στο αριστερό του υποδένδρο αριθμούς μικρότερους από τον ίδιο και στο δεξί του υποδένδρο αριθμούς μεγαλύτερους από τον ίδιο. Η ιδιότητα αυτή ισχύει τόσο για το αριστερό όσο και για το δεξί υποδένδρο.

Ορίστε την αναδρομική συνάρτηση:

```
Treeptr addtree(Treeptr p, int x)
```

Η συνάρτηση προσθέτει έναν αριθμό  $x$  στο ταξινομημένο δυαδικό δένδρο  $p$ , διατηρώντας το ταξινομημένο, και επιστρέφει το νέο δένδρο. Ο αλγόριθμος λειτουργεί ως εξής:

- Αν το δένδρο  $p$  είναι κενό (NULL), δημιουργείται ένας νέος κόμβος που περιέχει τον αριθμό  $x$ .
- Αν ο αριθμός που περιέχει ο τρέχων κόμβος είναι μεγαλύτερος από το  $x$ , η συνάρτηση καλείται για το αριστερό παιδί του κόμβου.
- Αν ο αριθμός είναι μικρότερος από το  $x$ , η συνάρτηση καλείται για το δεξί παιδί.
- Αν ο αριθμός είναι ίσος με το  $x$ , δεν γίνεται εισαγωγή.

Τροποποιήστε τη συνάρτηση `main` ώστε να διαβάζει αριθμούς από το πληκτρολόγιο μέχρι το τέλος της εισόδου και να τους προσθέτει στο δένδρο.

4.3 Κατασκευάστε την αναδρομική συνάρτηση:

```
void treeprint(Treeptr p);
```

Η συνάρτηση δέχεται ως όρισμα το δένδρο και εκτυπώνει τα περιεχόμενά του με in-order διάσχιση. Ο αλγόριθμος λειτουργεί ως εξής:

1. Αν το δένδρο είναι κενό (NULL), η συνάρτηση επιστρέφει.
2. Διαφορετικά, εκτελούνται τα εξής βήματα:
  - Καλείται η `treeprint` για το αριστερό παιδί.
  - Εκτυπώνεται η τιμή του τρέχοντος κόμβου.
  - Καλείται η `treeprint` για το δεξί παιδί.

Καλέστε τη `treeprint` από τη συνάρτηση `main` για να εκτυπώσετε το δένδρο που κατασκευάσατε.