

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΑΘΗΜΑ: ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ (ΕΡΓΑΣΤΗΡΙΟ)
ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 3

ΠΡΟΘΕΣΜΙΑ ΥΠΟΒΟΛΗΣ: **Κυριακή 25/01/2015**

ΘΕΜΑΤΟΛΟΓΙΑ: Binary Search Trees – Singly Linked Lists

Η παρούσα άσκηση αποτελεί συνέχεια των 2 προηγούμενων.

Έτσι, έχουμε και πάλι μια συλλογή βιβλίων και των συγγραφέων τους. Αυτή τη φορά τα δεδομένα αποθηκεύονται σε Binary Search Tree, ως εξής:

```
typedef struct Author{/*o kathe syggrafeas
    int author_id;
    char fname[50];
    char lname [50];
}Author;

typedef struct BookNode{/*o kombos toy BST
    int book_id;
    char title[50];
    char publisher [50];
    int numOfAuthors;
    Author authors[5];/*statikos pinakas me tous syggrafeis
    struct BookNode* left;/*aristero paidi
    struct BookNode* right;/*deksi paidi
    struct ListBookNode* ListPointer;
    /*to pedio ListPointer einai enas deikths ston kombo ths listas
    ston opoio enai apothikeymana ta idia stoixeia gia to kathe biblio.
}BookNode;

//olh h plhroforia apothgikeyetai se mia struct me 2 pedia
//1: enas pointer sth riza enos BST
//2: ton arithmo twv bibliwn
typedef struct Books{
    BookNode* root;
    Int numOfBooks;
} Books;
```

Επιπλέον, για να κρατάμε ένα backup των δεδομένων μας, έχουμε φτιάξει και μια απλά συνδεδεμένη λίστα (Singly Linked List) όπου αποθηκεύουμε τα ίδια δεδομένα, ως εξής:

```
typedef struct ListBookNode{/*o kombos ths listas
    int book_id;
    char title[50];
    char publisher [50];
    int numOfAuthors;
    Author authors[5];/*statikos pinakas me tous syggrafeis
    struct ListBookNode* next;/*pointer ston epomeno kombo
}ListBookNode;

typedef struct BooksBackUp{
    BookNode* head;
    Int numOfBooks;
} BooksBackUp;
```

Στη συνέχεια θα πρέπει να υλοποιήσουμε τις πιο κάτω λειτουργίες:

1. Ανάγνωση από 2 αρχεία, και «φόρτωμα» (populate) τόσο του BST, όσο και της συνδεδεμένης λίστας.
2. Γράψιμο σε 2 αρχεία (όπως η περίπτωση 1)
3. «Ενημέρωση» του αρχείου της λίστας (αρχείο-2) από το αρχείο του δένδρου (αρχείο-1). (αυτό θα υλοποιηθεί με «κοπιάρισμα» του ενός αρχείου στο άλλο)
4. Εύρεση στοιχείων βιβλίου βάσει book_id
5. Εύρεση στοιχείων συγγραφέα βάσει author_id
6. Εμφάνιση συγγραφέων ενός βιβλίου βάσει τίτλου
7. Εμφάνιση των βιβλίων που έχει γράψει ένας συγγραφέας (βάσει last_name)
8. Εισαγωγή βιβλίου
9. Εισαγωγή συγγραφέα
10. Διαγραφή βιβλίου
11. Εμφάνιση όλων των βιβλίων μαζί και των συγγραφέων του καθενός
12. Update της λίστας από τα στοιχεία του BST.

Παρατηρήσεις για την υλοποίηση.

Όλες οι πράξεις (αναζήτηση, εκτύπωση στην οθόνη κλπ) θα γίνονται στο BST.

Όταν γίνεται οποιαδήποτε αλλαγή στο BST αυτόματα θα πρέπει «ενημερώνεται» και η συνδεδεμένη λίστα.

Θα κρατάμε 2 αρχεία δεδομένων, ένα για το δένδρο και ένα για τη λίστα.

Προσπαθήστε να βρείτε έτοιμο κώδικα από το internet, για το χειρισμό του BST, κυρίως δε για τη διαγραφή κόμβου, που έχει μεγαλύτερη δυσκολία. Όχι όμως να είναι όμοιος με κώδικα κάποιου συμφοιτητή σας.

Το format των αρχείων θα είναι όπως και στις προηγούμενες ασκήσεις, με την πρώτη γραμμή να έχει το πλήθος των εγγραφών της βάσης δεδομένων.

Οι άσκηση είναι ατομική. Η ανταλλαγή απόψεων μεταξύ σας ενθαρρύνεται, όχι όμως και η ανταλλαγή-αντιγραφή κώδικα. Κώδικες με μεγάλο βαθμό ομοιότητας δεν θα βαθμολογούνται.

Για οποιαδήποτε απορία ή διευκρίνιση, θα «ανοίξει» φόρουμ στο eclass όπου μπορείτε να συμμετάσχετε, όχι όμως με παράθεση αυτούσιων κομματιών κώδικα.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!