

# Δομές Δεδομένων Εργαστηριακή Άσκηση 2012-2013

## Μέρος Α (30%)

Στην παρούσα εργαστηριακή άσκηση ζητείται να γίνει το λογισμικό για την οργάνωση μίας συλλογής βιβλίων. Η αποθήκευση όλης της πληροφορίας θα γίνεται σε ειδικά διαμορφωμένο αρχείο δεδομένων. Με την έναρξη του λογισμικού διαχείρισης, όλες οι εγγραφές από το αρχείο μεταφορτώνονται στην κύρια μνήμη όπου ο χρήστης θα μπορεί να επεξεργάζεται την συλλογή (πχ. να προσθέτει ένα βιβλίο, να διαγράφει, να επεξεργάζεται μία εγγραφή, να κάνει ερωτήματα αναζήτησης κλπ.). Με την λήξη του προγράμματος ή όταν το επιλέγει ο χρήστης, τα δεδομένα από την μνήμη θα αποθηκεύονται στο αρχείο. Παρακάτω περιγράφονται κάποιες λειτουργίες δίνοντας παραδείγματα σε C. Η υλοποίηση των λειτουργιών μπορεί να γίνει και με άλλες γλώσσες προγραμματισμού (πχ. C++, C#, Java, perl, python).

Για κάθε εγγραφή Βιβλίου χρησιμοποιείται η ακόλουθη δομή:

```
typedef struct Book{
int id;
char title[256];
char summary[2048];
    int numberOfAuthors;
    Author * authors;
};
```

Για κάθε Συγγραφέα χρησιμοποιείται η ακόλουθη δομή:

```
typedef struct Author{
    char firstName[56];
    char lastName[56];
};
```

Τα Βιβλία οργανώνονται στην ακόλουθη δομή:

```
typedef struct Books{
    Book *arr;
    int numberOfBooks;
};
```

Υλοποιήστε τις κατάλληλες συναρτήσεις ώστε να μπορούμε να πραγματοποιούμε τις ακόλουθες πράξεις μέσα από ένα μενού επιλογών:

1. Load books from file
2. Save books to file
3. Add a book
4. Delete a book by id
5. Display a book by id
6. Display a book by title
7. Display books
8. Display books by surname search
9. Exit

Το αρχείο στο οποίο θα σώζουμε ή θα διαβάζουμε θα περνιέται σαν argument στη main. Σε περίπτωση που δεν περαστεί argument, θα χρησιμοποιείται μια default τιμή για ένα τοπικό αρχείο. Στην επιλογή 1, ο τρέχον πίνακας arr θα αποδεσμεύεται και θα δημιουργείται εκ νέου με malloc. Για το λόγο αυτό, η πρώτη τιμή που θα γράφεται στο αρχείο θα είναι το πλήθος των εγγραφών (Βιβλίων) που ακολουθούν. Στην επιλογή 3, ο πίνακας θα αυξάνει σε μέγεθος κατά 1 (με realloc) και το νέο στοιχείο θα μπαίνει στο τέλος. Στις επιλογές 4 και 5, θα γίνεται γραμμική αναζήτηση χρησιμοποιώντας id, ενώ στις επιλογές 6 και 8, θα γίνεται αναζήτηση με βάση τον τίτλο και το επώνυμο αντίστοιχα. Στην επιλογή 4, για να διαγραφεί η εγγραφή αντιμετωπίζουμε με το τελευταίο στοιχείο του πίνακα και κατόπιν realloc σε -1 μέγεθος. **Σε αυτό το πρώτο ερώτημα οι πράξεις της αναζήτησης θα γίνονται με γραμμική αναζήτηση.**

**Μέρος Β (20%):** Κάντε την αναζήτηση με βάση το id με χρήση το δυαδικής αναζήτησης και με binary interpolation search. Αυτό έχει ως αποτέλεσμα να πρέπει να διατηρείται την συλλογή ταξινομημένη με βάση το id.

**Μέρος Γ (20%):** Με χρήση των ψηφιακών δέντρων (digital tries), οργανώστε την αναζήτηση με βάση των τίτλο και με βάση το επίθετο. Η κάθε δομή (1 για τους τίτλους και 1 για τα επίθετα) θα κατασκευάζεται μόλις φορτώνονται οι εγγραφές στη μνήμη και θα αποδεσμεύεται όταν τερματίζει η εφαρμογή.

**Μέρος Δ (20%):** Με χρήση των AVL, οργανώστε την αναζήτηση με βάση το id. Η δομή θα δημιουργείται όταν φορτώνονται οι εγγραφές στη μνήμη, και θα αποδεσμεύεται όταν τερματίζει η εφαρμογή.

**Μέρος Ε (10%):** Δημιουργήστε μία αρκετά μεγάλη συλλογή από Βιβλία  $\sim 100\text{Mb}$  (μπορείτε να πάρετε ένα υποσύνολο από το αρχείο <http://dblp.uni-trier.de/xml/>). Στη συνέχεια εκτελέστε ένα μεγάλο αριθμό από αναζητήσεις (πχ.  $>1000$ ) είτε με βάση το id, είτε με βάση κάποια επίθετα. Μετρήστε τον μέσο χρόνο αναζήτησης σε κάθε ένα από τα προηγούμενα μέρη. Τι παρατηρείτε όσον αφορά την απόδοση; Αναλύστε τα συμπεράσματά σας με βάση την απόδοση της κάθε μεθόδου που μας δίνει η θεωρία.

## Διαδικαστικά:

Η άσκηση είναι ατομική!

**Παράδοση: 30/06/2013**

Εξέταση Εργαστηριακής Άσκησης: Θα ανακοινωθεί μετά την παράδοση των ασκήσεων

Παράδοση μέσω της πλατφόρμας του eclass (Eclass- Δομές Δεδομένων) [eclass.upatras.gr](http://eclass.upatras.gr)

**Προσοχή:** στη μορφή και το όνομα του αρχείου που θα αποστείλετε (βλέπετε παρακάτω στα παραδοτέα). Ένα αρχείο rar/zip με όνομα αριθμόςμητρώου.zip (πχ 1821.zip)

Το αρχείο αυτό θα περιέχει :

- Αναφορά (html,doc,pdf) με τα ζητούμενα του κάθε ερωτήματος (τεκμηρίωση, γραφικές παραστάσεις κλπ.). Στην αρχή της αναφοράς να αναγράφεται επίσης το όνομα, το ΑΜ και το e-mail
- Τον πηγαίο κώδικα ΟΛΩΝ των προγραμμάτων που έχετε υλοποιήσει
- Αρχεία που βοηθούν στην μεταγλώττιση, πχ makefiles, solutions files του VS, κλπ.
- **Το αρχείο αυτό ΔΕΝ θα περιέχει τα εκτελέσιμα αρχεία**

Το μάθημα περιλαμβάνει τελική γραπτή εξέταση και εργαστηριακές ασκήσεις.

1. Η άσκηση είναι ατομική
2. Η παράδοση όλων των εργαστηριακών ασκήσεων είναι απαραίτητη για την συμμετοχή στην γραπτή εξέταση. Τα έτη  $\leq 4$ οι πρέπει να έχουν ολοκληρώσει τις εργαστηριακές ασκήσεις για να προσέλθουν σε γραπτή εξέταση. Οι βαθμοί των ασκήσεων θα προσμετρήσουν στην τελική βαθμολογία εφόσον ο βαθμός της τελικής γραπτής εξέτασης είναι  $\geq 5$ .
3. Βαθμολογία Άσκησης και Τελικός Βαθμός Μαθήματος:

Ανάλογα με το ποσοστό ολοκλήρωσης της εργασίας αυτή μπορεί να προσθέσει μέχρι και 1.5 βαθμό στον βαθμό της γραπτής εξέτασης εφόσον παραδοθεί στην τρέχουσα περίοδο (Εξεταστική Ιουνίου) ενώ αν παραδοθεί τον Σεπτέμβριο μπορεί να προσθέσει μέχρι και 1 βαθμό. Η βαθμολογία της εργασίας έχει ισχύ μόνο για το τρέχον ακ.έτος στο οποίο έχει παραδοθεί. Από τη στιγμή που έχει παραδοθεί μια εργασία θεωρείται ότι ο φοιτητής έχει καλύψει την υποχρέωση του ως προς αυτή και δεν χρειάζεται να παραδώσει ξανά σε επόμενα ακ. έτη όπου θα προσέλθει σε γραπτή εξέταση. Βεβαίως μπορεί να ξαναπαραδώσει σε περίπτωση που επιθυμεί να βελτιώσει τον βαθμό της γραπτής εξέτασης.

4. Απορίες σχετικά με τη θεωρία/φροντιστήριο/εργαστηριακές ασκήσεις υποβάλλονται στο φόρουμ του μαθήματος ([my.ceid](http://my.ceid))