



Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Μάθημα: Δομές Δεδομένων

Εργασία Ακ. Έτος 2014-2015

Μέρος Α (30%):

- Στην παρούσα εργασία ζητείται να γραφτεί το λογισμικό για την οργάνωση στοιχείων εταιριών.
- 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 μια εταιρία, να διαγράψει, να επεξεργαστεί 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 οι βασικές δομή των δεδομένων και λειτουργίες με παραδείγματα σε C.
- Η υλοποίηση των λειτουργιών πρέπει να γίνει με άλλη γλώσσα προγραμματισμού (πχ. C++, C#, Java, perl, python).

**Για κάθε εγγραφή Εταιρίας
χρησιμοποιείται η
ακόλουθη δομή:**

```
typedef struct Company {  
    int id;  
    char title[256];  
    char summary[2048];  
    int numberOfEmployees;  
    Employee *employees;  
};
```

**Για κάθε Εργαζόμενο
χρησιμοποιείται η
ακόλουθη δομή:**

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

**Οι Εταιρίες οργανώνονται
στην ακόλουθη δομή:**

```
typedef struct Companies{  
    Company *arr;  
    int numberOfCompanies;  
};
```

- Υλοποιήστε τις κατάλληλες συναρτήσεις ώστε να μπορούν να πραγματοποιηθούν οι ακόλουθες πράξεις μέσα από ένα μενού επιλογών:

1. Load companies from file
2. Save companies to file
3. Add a company
4. Delete a company by id
5. Display a company by id
6. Display companies
7. Display companies by surname search
8. Exit

- Το αρχείο στο οποίο θα σώζονται ή θα διαβάζονται τα δεδομένα θα περνιέται σαν argument στη main. Σε περίπτωση που δεν περαστεί argument, θα χρησιμοποιείται μια default τιμή για ένα τοπικό αρχείο.

- Προτείνεται το αρχείο να έχει την παρακάτω csv μορφή για να υπάρχει εύκολη επέκταση όταν θα χρησιμοποιηθεί το αρχείο εισόδου του 5^{ου} μέρους της εργασίας.

id; title; summary; numberOfEmployees; employee1; employee2; employee3;; employee7

- Στην επιλογή 1, ο τρέχων πίνακας arr 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 (Εταιριών) 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 στην επιλογή 7, θα γίνεται αναζήτηση με βάση το επώνυμο αντίστοιχα. Στην επιλογή 4, για να διαγραφεί η εγγραφή αντιμετωθούμε με το τελευταίο στοιχείο του πίνακα και κατόπιν realloc σε -1 μέγεθος.
- Σε αυτό το πρώτο μέρος οι πράξεις της αναζήτησης θα γίνονται με γραμμική αναζήτηση.

Μέρος Β (20%):

- **Υλοποιείτε** την αναζήτηση με βάση το id με χρήση δυαδικής αναζήτησης και binary interpolation search. Αυτό έχει ως αποτέλεσμα να πρέπει να διατηρείται η συλλογή ταξινομημένη με βάση το id.

Μέρος Γ (30%):

- 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. Πρέπει να **υλοποιήσετε** τη δομή τις πράξεις εισαγωγής, διαγραφής και εύρεσης στο AVL.

Μέρος Δ (10%):

- Με χρήση των ψηφιακών δέντρων (digital tries), οργανώστε την αναζήτηση με βάση το επίθετο. Η δομή θα κατασκευάζεται μόλις φορτώνονται οι εγγραφές στη μνήμη και θα αποδεσμεύεται όταν τερματίζει η εφαρμογή. Σε περίπτωση που υπάρχει ξανά το επίθετο του εργαζομένου από άλλη εταιρία πρέπει στο αποτέλεσμα να επιστρέφονται όλες οι πιθανές εταιρίες. Πρέπει να **υλοποιήσετε** τη δομή και τις πράξεις εισαγωγής και εύρεσης στο TRIE.

Μέρος Ε (10%):

- Χρησιμοποιείτε το αρχείο data.csv (; delimited) που σας δίνεται με 10000 εγγραφές και εκτελέστε ένα μεγάλο αριθμό από αναζητήσεις (πχ. >1000) με βάση το id και με βάση κάποια επίθετα. Μετρήστε τον μέσο χρόνο αναζήτησης σε συγκρίσεις και εκτελέσιμο χρόνο σε κάθε ένα από τα προηγούμενα μέρη. Τι παρατηρείτε όσον αφορά την απόδοση; Αναλύστε τα συμπεράσματα σας με βάση την απόδοση της κάθε μεθόδου που μας δίνει η θεωρία. Δώστε γραφήματα που να δείχνουν τα αποτελέσματα σας. Τα πειράματα πρέπει να εκτελεστούν με τυχαία σύνολα ερωτημάτων ίδιου μεγέθους τουλάχιστον 10 φορές και παίρνετε ως αποτέλεσμα τον μέσο όρο.
- **Προσοχή!** Το αρχείο εισόδου περιέχει σε όλες τις γραμμές 7 ονόματα υπαλλήλων. Θα παίρνετε σαν είσοδο μόνο τα πρώτα x που λέει ο αριθμός υπαλλήλων (numberOfEmployees).

Σημείωση 1: Το λογισμικό σας πρέπει να τρέχει και ανεξαρτήτως του αρχείου εισόδου του 5^{ου} μέρους. Θα παίρνει ως είσοδο ένα αρχείο data.csv που αν είναι άδειο δεν θα φορτώνει κάτι στη μνήμη και θα αποθηκεύονται σε αυτό όλες οι εισαγωγές του χρήστη. Αν περιέχει εγγραφές αφού φορτωθούν και γίνουν αλλαγές θα αποθηκεύονται πίσω στο ίδιο αρχείο. Οι εγγραφές στο αρχείο δεν είναι ανάγκη να είναι σε αύξουσα ταξινόμηση με βάση το id αλλά όταν η μέθοδος που εκτελείται επιβάλει τα ids να είναι σε αύξουσα σειρά πρέπει να γίνονται οι κατάλληλες ενέργειες.

Σημείωση 2: Στο μενού ανάλογα με τη βασική επιλογή μπορείτε να βάλετε υποεπιλογές ώστε να επιλέγεται η μέθοδος της αναζήτησης.

Παραδοχή: Τα ids είναι μοναδικά και δεν επιλαμβάνονται. Δεν χρειάζεται αν δεν θέλετε να γίνεται έλεγχος αλλά να είστε σίγουροι στις δοκιμές σας ότι δεν επαναλαμβάνονται.

Παρότρυνση: Μπορείτε να χρησιμοποιήσετε ότι προσφέρουν οι βιβλιοθήκες που παρουσιάστηκαν στο μάθημα αλλά **όχι** τυχόν έτοιμες υλοποιήσεις των αναζητήσεων, των AVL και των digital tries.

Διαδικαστικά:

Η άσκηση είναι ατομική!

Παράδοση: 28/06/2015

Εξέταση Εργαστηριακής Άσκησης: Θα ανακοινωθεί μετά την παράδοση των ασκήσεων

Παράδοση μέσω της πλατφόρμας του eclass (Eclass- Δομές Δεδομένων) 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)

Πρέπει να χρησιμοποιηθεί **ίδια γλώσσα προγραμματισμού** για όλα τα ζητούμενα.

Το αρχείο αυτό θα περιέχει:

- Αναφορά (html, doc, pdf) με τα ζητούμενα του κάθε ερωτήματος (τεκμηρίωση, γραφικές παραστάσεις κλπ). Στην αρχή της αναφοράς να αναγράφετε επίσης το όνομα, το ΑΜ, το e-mail και η γλώσσα προγραμματισμού που χρησιμοποιήσατε.
- 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 εργασία.

1. Η παράδοση όλων των εργαστηριακών ασκήσεων είναι απαραίτητη για την συμμετοχή στην γραπτή εξέταση. Τα έτη ≤ 4 ου πρέπει να έχουν 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 ≥ 5 .

2. Βαθμολογία Άσκησης και Τελικός Βαθμός Μαθήματος:

Ανάλογα με το ποσοστό ολοκλήρωσης της εργασίας, και όταν αυτό είναι ικανοποιητικό, 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. Θεωρείται ότι έχει παραδοθεί εργασία αν ο προσθετικός βαθμός που ανακοινωθεί είναι μεγαλύτερος του 0 (0-1,5) και έχει υλοποιήσει $\geq 50\%$ της εργασίας. 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.

3. Απορίες σχετικά με τις εργασίες υποβάλλονται στο φόρουμ του μαθήματος (my.ceid) και στις ώρες γραφείου που θα ανακοινωθούν με βάση το ΑΜ σας.