

**ΣΕΤ ΑΣΚΗΣΕΩΝ 4****ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2011-2012**

**Προθεσμία: 8/1/12, 22:00**

**Περιεχόμενα**

- [Διαβάστε πριν ξεκινήσετε](#)
- [Εκφώνηση άσκησης](#)
- [Οδηγίες αποστολής άσκησης](#)

**Πριν ξεκινήσετε (ΔΙΑΒΑΣΤΕ ΑΥΤΗ ΤΗΝ ΕΝΟΤΗΤΑ!!)**

Διαβάστε *ολόκληρη* την εκφώνηση προσεκτικά και “σχεδιάστε” το πρόγραμμά σας στο χαρτί.

Αποθηκεύστε το πρόγραμμά σας σε ένα αρχείο με όνομα hw4 .c

Μην προσπαθήσετε να γράψετε το πρόγραμμα μια κι έξω. Να υλοποιείτε μία-μία τις συναρτήσεις με τη μέθοδο που χρησιμοποιήσαμε στο φροντιστήριο (αρχικά όλες οι συναρτήσεις είναι άδειες). Μετά την υλοποίηση κάθε μίας συνάρτησης, την τεστάρετε εκτεταμένα, διορθώνετε ότι λάθη έχει και μετά προχωράτε στην επόμενη συνάρτηση. Μην παραλείπετε να αποθηκεύετε αντίγραφα του προγράμματός σας κάθε φορά που υλοποιείτε ένα μεγάλο κομμάτι, ώστε αν σε κάποιο επόμενο στάδιο καταστραφεί το αρχείο σας, θα έχετε μια παλιότερη έκδοση από την οποία μπορείτε να συνεχίσετε. Τις επιμέρους “εκδόσεις” να τις ονομάζετε με διαφορετικά ονόματα.

Η άσκηση βαθμολογείται ως προς την ορθότητα, καλή χρήση συναρτήσεων, σωστή χρήση σταθερών, αναγνωσιμότητα και σχολιασμό. Επίσης, πρέπει να ακολουθήσετε ακριβώς τις προδιαγραφές κάθε συνάρτησης και η έξοδος του προγράμματός σας να είναι ακριβώς ίδια με την αναμενόμενη.

Μη διστάζετε να ζητήσετε βοήθεια! Μπορείτε να χρησιμοποιήσετε το forum προγραμματισμού (<http://inf-server.inf.uth.gr/courses/coding/>) και φυσικά email.

Η εργασία αυτή μπορεί να γίνει σε ομάδες μέχρι 2 ατόμων. Μπορείτε να συζητάτε τις ασκήσεις με συμμαθητές σας αλλά δεν επιτρέπεται η ανταλλαγή κώδικα με οποιοδήποτε τρόπο.

**Ξεκινήστε νωρίς!** Ο προγραμματισμός είναι πάντα ΠΟΛΥ πιο χρονοβόρος από ό,τι περιμένετε.

Εκπρόθεσμες ασκήσεις δε γίνονται δεκτές.

## Άσκηση

### Εισαγωγή

Θα γράψετε ένα πρόγραμμα το οποίο διαβάζει από ένα αρχείο πληροφορίες (τίτλο και τιμή) για ένα άγνωστο αριθμό βιβλίων, τις αποθηκεύει σε μια διασυνδεδεμένη δομή η οποία διευκολύνει την ταξινόμησή τους, και τέλος εκτυπώνει σε δύο διαφορετικά αρχεία τις πληροφορίες ταξινομημένες με βάση τον τίτλο και με βάση την τιμή αντίστοιχα.

Ακολουθούν λεπτομερείς οδηγίες για το πώς πρέπει να λειτουργεί το πρόγραμμά σας και τι συναρτήσεις πρέπει να περιέχει.

### Δομές Δεδομένων

Θα χρησιμοποιήσουμε δύο πίνακες από δείκτες σε λίστες με κοινά σημεία. Για την ακρίβεια:

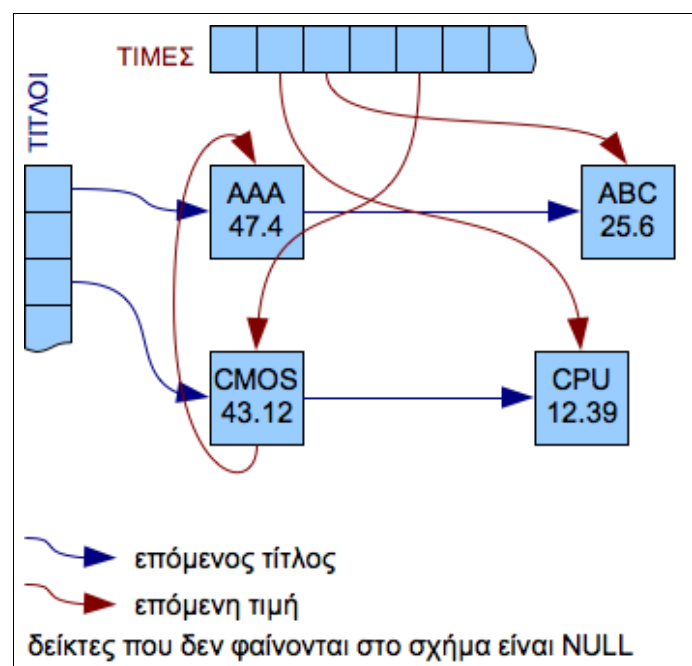
Γνωρίζουμε πως οι τιμές των βιβλίων κυμαίνονται από 0 (μη συμπεριλαμβανομένου) έως και 100. Θέλουμε να κατασκευάσουμε ένα πίνακα από δείκτες προς την αρχή λιστών, έτσι ώστε στην 1η θέση του πίνακα να βρίσκεται δείκτης προς την αρχή λίστας που περιέχει όλα τα βιβλία με τιμή το πολύ 10, στη δεύτερη θέση να βρίσκεται δείκτης προς λίστα που περιέχει όλα τα βιβλία με τιμή μεγαλύτερη του 10 αλλά το πολύ 20, κ.ο.κ. Γενικά στη θέση  $i$  θα βρίσκεται μια λίστα που περιέχει όλα τα βιβλία με τιμή μεγαλύτερη από  $(i-1)*10$  και έως το πολύ  $i*10$ . Τα περιεχόμενα κάθε λίστας διατηρούνται ταξινομημένα σε αύξουσα σειρά.

Παράλληλα, γνωρίζουμε πως οι τίτλοι των βιβλίων ξεκινούν από ένα γράμμα μεταξύ 'Α' και 'Ζ' (στο αγγλικό αλφάβητο). Θέλουμε να κατασκευάσουμε ένα πίνακα από δείκτες σε αρχή λιστών έτσι ώστε στην 1η θέση του πίνακα να βρίσκεται δείκτης προς μια λίστα που περιέχει όλα τα βιβλία με τίτλους που αρχίζουν από 'Α', στη 2η θέση του πίνακα να βρίσκεται δείκτης προς μια λίστα που περιέχει όλα τα βιβλία με τίτλους που αρχίζουν από 'Β', κ.ο.κ. Τα περιεχόμενα καθεμιάς από αυτές τις λίστες διατηρούνται ταξινομημένα σε λεξικογραφικά αύξουσα σειρά.

Για κάθε βιβλίο θα υπάρχει **ένας μόνο κόμβος** ο οποίος όμως θα συμμετέχει σε δύο λίστες. Για παράδειγμα, αν έχουμε βιβλία με τους παρακάτω τίτλους και τιμές

ABC 25.6  
CPU 12.39  
AAA 47.4  
CMOS 43.12

τότε θα είναι οργανωμένα όπως φαίνεται στο παρακάτω σχήμα:



Οι λίστες είναι απλά διασυνδεδεμένες, μη-κυκλικές, χωρίς τερματικό. Όπως βλέπετε στο σχήμα, κάθε κόμβος έχει δύο "επόμενους" δείκτες. Έναν προς τον επόμενο κόμβο με βάση τον τίτλο (στην ίδια "σειρά" στο σχήμα) κι έναν προς τον επόμενο κόμβο με βάση την τιμή (στην ίδια "στήλη" στο σχήμα).

Επομένως, το struct που πρέπει να κατασκευάσετε για να αναπαραστήσετε ένα κόμβο θα περιέχει πεδία για:

- Τον τίτλο του βιβλίου (συμβολοσειρά μέχρι 50 χαρακτήρες, συμπεριλαμβανομένου του '\0')
- Την τιμή του βιβλίου (αριθμός κινητής υποδιαστολής μεγαλύτερος του 0 και το πολύ 100)
- Ένα δείκτη προς τον επόμενο κόμβο ως προς την τιμή
- Ένα δείκτη προς τον επόμενο κόμβο ως προς τον τίτλο

Τέλος, θα χρειαστεί να δηλώσετε ένα πίνακα δεικτών, μεγέθους 26 (πλήθος γραμμάτων αλφαβήτας), για την οργάνωση των δεδομένων με βάση τους τίτλους, κι έναν πίνακα δεικτών, μεγέθους 10 (πλήθος δεκάδων), για την οργάνωση των δεδομένων με βάση τις τιμές.

### **Στάδια προγράμματος**

Το πρόγραμμά σας πρέπει να αποτελείται από τα παρακάτω στάδια, τα οποία θα αναλυθούν σε επόμενες ενότητες:

1. Διαβάζει από τη γραμμή εντολής (ως όρισμα) το όνομα ενός αρχείου εισόδου και κατασκευάζει τα ονόματα δύο αρχείων εξόδου.
2. Διαβάζει τα στοιχεία των βιβλίων από το αρχείο εισόδου και τα αποθηκεύει στην παραπάνω δομή
3. Εκτυπώνει τα στοιχεία στα αρχεία εξόδου όπως περιγράφεται πιο κάτω
4. Απελευθερώνει όλη τη δυναμικά δεσμευμένη μνήμη.

### **1: Αρχείου εισόδου**

Διαβάστε το όνομα του αρχείου εισόδου ως όρισμα από τη γραμμή εντολής. Θα χρειαστεί να κατασκευάσετε δύο αρχεία εξόδου. Το ένα έχει όνομα ίδιο με του αρχείου εισόδου με κατάληξη .prices και το άλλο έχει όνομα ίδιο με του αρχείου εισόδου με κατάληξη .titles. Τα ονόματα των αρχείων εξόδου αποθηκεύστε τα σε κατάλληλες συμβολοσειρές για τις οποίες έχετε δεσμεύσει ακριβώς όση μνήμη χρειάζεται.

### **2: Ανάγνωση δεδομένων και εισαγωγή στη δομή**

Το αρχείο εισόδου περιέχει τα στοιχεία για ένα άγνωστο αριθμό βιβλίων. Σε κάθε μία γραμμή βρίσκεται ο τίτλος ενός βιβλίου (χωρίς κενά ανάμεσα στις λέξεις), κενό και η τιμή. Κάθε φορά που διαβάζετε ένα τίτλο και μια τιμή πρέπει να χρησιμοποιείτε μια συνάρτηση με όνομα insert για να κατασκευάσετε τον αντίστοιχο κόμβο και τον εισάγετε στη δομή. Η insert παίρνει τέσσερις παραμέτρους:

- τον πίνακα από λίστες όπου οι κόμβοι συνδέονται με βάση τον τίτλο
- τον πίνακα από λίστες όπου οι κόμβοι συνδέονται με βάση την τιμή
- τον τίτλο
- την τιμή

και δεν επιστρέφει τίποτα. Αν η εισαγωγή έχει γίνει σωστά, τότε τα βιβλία θα είναι οργανωμένα με τέτοιο τρόπο ώστε να είναι ταξινομημένα και ως προς τους τίτλους και ως προς τις τιμές.

### **3: Εκτύπωση στα αρχεία εξόδου**

Θυμηθείτε πως στο στάδιο 1 κατασκευάσατε δύο αρχεία εξόδου. Σε αυτό με επέκταση .price θέλουμε να γράψουμε τα στοιχεία των βιβλίων έτσι ώστε να είναι ταξινομημένα ως προς την τιμή, σε αύξουσα σειρά. Σε κάθε γραμμή γράφονται τα στοιχεία ενός βιβλίου ως εξής:

τίτλος      κενό      αριστερή\_παρένθεση      τιμή      δεξιά\_παρένθεση      αλλαγή\_γραμμής

Αντίστοιχα, στο αρχείο με επέκταση .title θέλουμε να γράψουμε τα στοιχεία των βιβλίων έτσι ώστε να είναι ταξινομημένα ως προς τον τίτλο, σε λεξικογραφικά αύξουσα σειρά. Σε κάθε γραμμή γράφονται τα στοιχεία ενός βιβλίου ως εξής:

τιμή      κενό      αριστερή\_παρένθεση      τίτλος      δεξιά\_παρένθεση      αλλαγή\_γραμμής

Αυτό θα το κάνετε χρησιμοποιώντας μια μόνο συνάρτηση την οποία θα καλέσετε δύο φορές. Η συνάρτηση θα έχει όνομα save και παίρνει τις παρακάτω παραμέτρους:

- μια τιμή που εκφράζει επιλογή ανάμεσα σε εκτύπωση ανά τίτλο ή ανά τιμή
- ένα δείκτη σε αρχείο (file descriptor)
- ένα πίνακα δεικτών σε κόμβους όπως αυτοί περιγράφονται πιο πάνω
- το μέγεθος του πίνακα (ακέραιος)

Την πρώτη φορά που θα καλέσετε τη save θα της δώσετε την επιλογή που σημαίνει 'εκτύπωση ως προς τίτλο', το αρχείο με επέκταση .title, τον αντίστοιχο πίνακα και το μέγεθός του. Η εγγραφή στο αρχείο θα γίνει όπως περιγράφεται πιο πάνω.

Τη δεύτερη φορά που θα καλέσετε τη save θα δώσετε τις αντίστοιχες παραμέτρους ώστε να γίνει η εγγραφή των στοιχείων ως προς την τιμή.

### **4: Απελευθέρωση δυναμικά δεσμευμένης μνήμης**

Γράψτε μια κατάλληλη συνάρτηση για την απελευθέρωση όλης της δυναμικά δεσμευμένης μνήμης του προγράμματος.

### **Υποθέσεις**

- Η είσοδος είναι πάντα σωστή. Για παράδειγμα, οι τιμές των βιβλίων θα είναι πάντα μεταξύ 0 (μη συμπεριλαμβανομένου) και 100 (συμπεριλαμβανομένου) επομένως δε χρειάζεται να κάνετε ελέγχους
- Οι τίτλοι των βιβλίων έχουν μέγεθος το πολύ 50 (μαζί με το '\0') και όλα τα γράμματα είναι κεφαλαία
- Οι τίτλοι των βιβλίων είναι μοναδικοί. Ομοίως οι τιμές.

### **Απαιτήσεις**

- Απαγορεύεται η χρήση καθολικών μεταβλητών
- Πρέπει να ελέγχετε πάντα τις περιπτώσεις λάθους κατά την εκτέλεση (πχ αποτυχία δέσμευσης μνήμης, ανοίγματος αρχείου, κτλ), να εκτυπώνετε βοηθητικό μήνυμα και να χειρίζεστε το πρόβλημα με κατάλληλο τρόπο
- Για τη διαχείριση συμβολοσειρών πρέπει να χρησιμοποιήσετε συναρτήσεις από τη string.h

**Πριν παραδώσετε το πρόγραμμά σας, προσθέστε σε σχόλια στην αρχή του αρχείου τα πλήρη ονόματα και ΑΜ των μελών της ομάδας. Παρακαλούμε να γράφετε τα σχόλια ΜΟΝΟ με λατινικούς χαρακτήρες.**

**Αρχεία προς παράδοση: hw4.c**

## **Πώς να παραδώσετε τη δουλειά σας**

**(Ακολουθείστε τις οδηγίες ακριβώς αλλιώς μπορεί να μη δούμε τα αρχεία σας)**

Κατασκευάστε ένα φάκελο με όνομα `ερwnumο1_AM1_ερwnumο2_AM2` και αντιγράψτε μέσα σε αυτόν το `hw4.c`

Πηγαίnete στο φάκελο μέσα στον οποίο βρίσκεται το `ερwnumο1_AM1_ερwnumο2_AM2` που κατασκευάσατε και γράψτε την παρακάτω εντολή:

```
tar czf ερwnumο1_AM1_ερwnumο2_AM2.tar.gz ερwnumο1_AM1_ερwnumο2_AM2
```

Στείλτε email:

- στη διεύθυνση **`ce120lab@gmail.com`**
- αντίγραφο (CC) στον άλλο μέλος της ομάδας σας
- θέμα (subject) **CE120 hw4**
- και επικολλημένο αρχείο το `ερwnumο1_AM1_ερwnumο2_AM2.tar.gz`