

I. STRUCTS: ΔΟΜΕΣ ΚΑΙ ΠΙΝΑΚΕΣ

Προκειμένου να αναβαθμίσετε το πρόγραμμα βαθμολογίου για μία τάξη Λυκείου, θα πρέπει να μετατρέψετε τα υπάρχοντα δεδομένα σε δομές (struct), **μία για κάθε μαθητή**. Συγκεκριμένα, σας δίνονται οι πίνακες `onoma`, `am` και `vathmos`, οι οποίοι έχουν, αντίστοιχα, τα ονόματα, τους αριθμούς μητρώου και τη βαθμολογία των μαθητών.

Αυτό που σας ζητείται είναι:

1. Να αποθηκεύσετε τα στοιχεία χρησιμοποιώντας μία δομή (struct) ανά μαθητή.
2. Να τυπώσετε τα περιεχόμενα των δομών.
3. Με χρήση των δομών, να υπολογίσετε τον μέσο βαθμό.
4. Με χρήση των δομών, να τυπώσετε τα ονόματα των μαθητών που κόπηκαν (δηλαδή είχαν βαθμολογία κάτω του 10).
5. Να αλλάξουν τα ονόματα όσων κόπηκαν ώστε το τέλος να υπάρχει το «(failed)». Π.χ. ο «Vasilios Vasiliou» να γίνει «Vasilios Vasiliou (failed)».

Υλοποιήστε τα ζητούμενα χρησιμοποιώντας **έναν πίνακα από δομές**. Θα πρέπει να δηλώσετε τη δομή με τα πεδία της, να δεσμεύσετε δυναμικά τον πίνακα από δομές και να αντιγράψετε στον πίνακα αυτόν τα στοιχεία από τους 3 πίνακες που σας δίνονται. Τέλος θα πρέπει να κάνετε και τις αντίστοιχες λειτουργίες χρησιμοποιώντας τα υποδείγματα των συναρτήσεων όπως σας δίνονται στο αρχείο `skell1.c`.

II. ΑΡΧΕΙΑ ΚΕΙΜΕΝΟΥ

Στο αρχείο `data.txt` υπάρχει μια σειρά από ακεραίους αριθμούς. Ο πρώτος από αυτούς τους αριθμούς καθορίζει πόσοι ακόμα αριθμοί υπάρχουν στο αρχείο. Αν για παράδειγμα αυτός ο αριθμός είναι ο 4, τότε ακολουθούν άλλοι 4 ακέραιοι αριθμοί στο αρχείο.

- α) Να γραφεί πρόγραμμα το οποίο ανοίγει αυτό το αρχείο, διαβάζει τα δεδομένα και τα αποθηκεύει σε έναν κατάλληλο πίνακα ακεραίων που θα δημιουργηθεί δυναμικά με την χρήση της `malloc()`. Στην συνέχεια το πρόγραμμα θα πρέπει να αποθηκεύσει τα δεδομένα που διάβασε σε δύο διαφορετικά αρχεία: το αρχείο `positive.txt` και το αρχείο `negative.txt`. Στο αρχείο `positive.txt` θα αποθηκευτούν οι θετικοί αριθμοί από αυτούς που διάβασε και στο αρχείο `negative.txt` θα αποθηκευτούν οι αρνητικοί αριθμοί. Τόσο το αρχείο `positive.txt` όσο και το αρχείο `negative.txt` να διατηρήσουν τη μορφοποίηση του αρχικού αρχείου, δηλαδή στην πρώτη γραμμή κάθε αρχείου θα πρέπει να υπάρχει το πλήθος των αριθμών που ακολουθούν.
- β) Να προστεθεί κώδικας ο οποίος επαναφέρει τη θέση του αρχείου `positive.txt` στην αρχή, ξαναδιαβάζει τους αριθμούς από το αρχείο `positive.txt` και υπολογίζει το άθροισμά τους.

III. STRUCTS + ΑΡΧΕΙΑ

Θα πρέπει το βαθμολόγιο του Προβλήματος I αρχικά να δίνεται από τον χρήστη και από εκεί και μετά να φυλάσσεται σε αρχείο. Συγκεκριμένα:

1. Κατά την **έξοδο** του προγράμματός σας, τον πίνακα από δομές τον αποθηκεύετε σε ένα αρχείο κειμένου `students.txt` (αποθηκεύετε ένα-ένα τα πεδία, σε ξεχωριστές γραμμές, με `fprintf()`). Το πρώτο πράγμα που πρέπει να αποθηκεύσετε είναι το πλήθος των μαθητών.
2. Κατά την **εκκίνηση** του προγράμματός σας, πρέπει να ελέγχετε αν υπάρχει το αρχείο `students.txt`:
 - Αν δεν υπάρχει, θα πρέπει τα διαβάσετε τα στοιχεία και τους βαθμούς των μαθητών από το πληκτρολόγιο και να τα βάλετε σε έναν πίνακα από δομές όπως στο Πρόβλημα I.
 - Αν υπάρχει, θα πρέπει να το διαβάσετε και να φτιάξετε τον πίνακα δομών από τα περιεχόμενά του αρχείου (θυμηθείτε ότι στην αρχή του αρχείου υπάρχει το πλήθος των μαθητών).Επομένως, ο πίνακας δομών σχηματίζεται είτε από το αρχείο (αν υπάρχει) είτε από το πληκτρολόγιο.

IV. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ: ΔΟΜΕΣ ΚΑΙ ΔΥΝΑΜΙΚΕΣ ΛΙΣΤΕΣ

Υλοποιήστε τα ζητούμενα του Προβλήματος I χρησιμοποιώντας αυτή τη φορά μία **λίστα από δομές** (αντί για πίνακα από δομές) στις οποίες θα αντιγράψετε τα στοιχεία από τους 3 πίνακες που σας δίνονται.

Θα πρέπει να δηλώσετε τη δομή και να συμπεριλάβετε και ένα πεδίο `next`, το οποίο θα δείχνει στον επόμενο κόμβο της λίστας. Επίσης, θα πρέπει να δηλώσετε και μία μεταβλητή `head`, η οποία θα είναι δείκτης με αρχική τιμή `NULL` (άδεια λίστα) και στη συνέχεια θα δείχνει πάντα στη δομή που μπήκε πιο πρόσφατα στη λίστα. Κάθε κόμβος της λίστας θα πρέπει να δεσμεύεται δυναμικά και θα εισάγεται στο σημείο που δείχνει η μεταβλητή `head`.

Τέλος θα πρέπει να κάνετε και τις αντίστοιχες λειτουργίες χρησιμοποιώντας τα υποδείγματα των συναρτήσεων όπως σας δίνονται στο αρχείο `skel2.c`.

V. ΕΠΙΠΛΕΟΝ ΕΞΑΣΚΗΣΗ: ΕΙΣΑΓΩΓΗ ΣΤΟ ΤΕΛΟΣ ΤΗΣ ΛΙΣΤΑΣ

Προσπαθήστε να επαναλάβετε το Πρόβλημα IV, αλλά αυτή τη φορά κάθε νέος κόμβος να εισάγεται στο τέλος και όχι στην αρχή της λίστας.