

# lab10: Structs

## Προετοιμασία

- Πλοηγηθείτε στον κατάλογο **ce120** και κατασκευάστε μέσα σε αυτόν ένα κατάλογο **lab10**.
- Κατεβάστε τα αρχεία ελέγχου **lab10.tar.gz** και αποθηκεύστε τα μέσα στον κατάλογο **lab10**.

## Οδηγίες αποστολής

Μέσα στον κατάλογο **lab10** κατασκευάστε ένα κατάλογο με όνομα

**lab10\_επώνυμο1\_AEM1\_επώνυμο2\_AEM2**

και αντιγράψτε σε αυτόν το αρχείο **lab10a.c** (εάν προλάβετε να ολοκληρώσετε και την προαιρετική άσκηση αντιγράψτε και το **lab10b.c** )

Συμπιέστε τον κατάλογο με χρήση της εντολής tar:

```
tar czvf lab10_επώνυμο1_AEM1_επώνυμο2_AEM2.tar.gz lab10_επώνυμο1_AEM1_επώνυμο2_AEM2
```

Το ένα μέλος της ομάδας μπαίνει στο λογαριασμό του στο eclass, πηγαίνει στη σημερινή εργασία, και αφού επιλέξει τη ΣΩΣΤΗ ομάδα, ανεβάζει την άσκηση και πατά **αποθήκευση**.

**Μην παραλείψετε να επιβεβαιώσετε ότι στείλατε το σωστό αρχείο!**

**ΑΠΑΓΟΡΕΥΕΤΑΙ Η ΧΡΗΣΗ GOTO, GLOBAL/STATIC ΜΕΤΑΒΛΗΤΩΝ ΚΑΙ gets**

**TIP: Χρησιμοποιήστε const σε όσες παραμέτρους ενδείκνυται.**

# Άσκηση 1

Κατεβάστε το αρχείο με όνομα **lab10a.c** και αποθηκεύστε το στον φάκελο **lab10**.

Σε αυτή την άσκηση σας δίνεται ο σκελετός ενός προγράμματος το οποίο καταχωρεί, αναζητεί και διαγράφει τα στοιχεία φοιτητών για το μάθημα του Προγραμματισμού. Τα στοιχεία ενός φοιτητή αναπαρίστανται από το `struct student` που σας δίνεται έτοιμο και περιέχει τις παρακάτω πληροφορίες:

- **Όνοματεπώνυμο:** Αναπαρίσταται με λατινικούς χαρακτήρες και δεν περιέχει κενό χαρακτήρα. Μέγιστο μέγεθος `MAX_NAME_SIZE` συμπεριλαμβανομένου του `'\0'`
- **AEM:** Θετικός ακέραιος.
- **Βαθμοί Quiz:** Πίνακας από πραγματικούς αριθμούς μεγέθους 2. Καταχωρεί τη βαθμολογία για τα 2 πρώτα Quiz. Λαμβάνει τιμές στο διάστημα `[0-10]`.
- **Τελική εξέταση:** Πραγματικός αριθμός. Λαμβάνει τιμές στο διάστημα `[0-10]`.

Το κυρίως πρόγραμμα υποθέτει ότι τα στοιχεία των φοιτητών καταχωρούνται σε ένα πίνακα από `struct student`, ενδεικτικού μεγέθους `MAX_STUDENTS`. **Καλείστε να τον δηλώσετε εσείς.**

Το πρόγραμμα εμφανίζει ένα μενού προς τον χρήστη (σας δίνεται έτοιμο) και τον καλεί να επιλέξει τα εξής: **α)** την προσθήκη εγγραφής, **β)** την εκτύπωση του συνόλου των εγγραφών που είναι αποθηκευμένες στον πίνακα **γ)** τη διαγραφή όλων των εγγραφών που περιέχουν το ίδιο όνομα φοιτητή ή **δ)** την αναζήτηση εγγραφής.

**Βήμα 1ο:** Γράψτε μία συνάρτηση για την εισαγωγή των στοιχείων ενός φοιτητή στον πίνακα. Η συνάρτηση λαμβάνει ως είσοδο **α)** ένα πίνακα από `struct student` και **β)** τη διεύθυνση ενός ακεραίου που αντιπροσωπεύει το πλήθος εγγραφών που έχουν γεμίσει στον πίνακα. Σημειώστε ότι οι γεμάτες θέσεις του πίνακα είναι συνεχόμενες (χωρίς κενά) ξεκινώντας από την αρχή του. Η συνάρτηση δεν επιστρέφει κάτι.

Η συνάρτηση ελέγχει εάν υπάρχουν κενές θέσεις στον πίνακα και εάν δεν υπάρχουν εμφανίζει το μήνυμα **"No space!"** ακολουθούμενο από χαρακτήρα αλλαγής γραμμής.

Διαφορετικά, εκτυπώνει το μήνυμα **"Enter student info (format: Name AEM Q1 Q2 final): "** και διαβάζει το όνομα, το AEM (θετικός ακέραιος) τους βαθμούς των Quiz και τον βαθμό της τελικής εξέτασης στη μορφή που περιγράφεται στο παραπάνω μήνυμα (διαχωριζόμενα μεταξύ τους με ένα λευκό χαρακτήρα).

Η συνάρτηση **ΧΩΡΙΣ** να κάνει έλεγχο εάν οι τιμές που δόθηκαν για το AEM και τις βαθμολογίες είναι αποδεκτές, καταχωρεί την πληροφορία που διάβασε στην επόμενη κενή θέση του πίνακα και αυξάνει το πλήθος εγγραφών του πίνακα κατά ένα μέσω της 2ης παραμέτρου.

**Καλέστε τη συνάρτηση** στο αντίστοιχο τμήμα της `main` έτσι ώστε κάθε φορά που ο χρήστης επιλέγει τη λειτουργία `append` να καλείται η συγκεκριμένη συνάρτηση και να προστίθεται μία εγγραφή στον πίνακα.

**Βήμα 2ο:** Σας δίνεται έτοιμη (**ΔΕΝ ΤΗΝ ΑΛΛΑΖΕΤΕ**) η συνάρτηση `print_student_info` η οποία λαμβάνει ως παράμετρο μία μεταβλητή τύπου `struct student` και εκτυπώνει τα περιεχόμενα της.

**Χρησιμοποιήστε** τη συνάρτηση αυτή, ώστε κάθε φορά που ο χρήστης επιλέγει τη λειτουργία `print` να εκτυπώνονται όλα τα περιεχόμενα που είναι αποθηκευμένα στον πίνακα και μόνον αυτά.

**Βήμα 3ο:** Γράψτε μία συνάρτηση διαγραφής στοιχείων η οποία λαμβάνει ως παραμέτρους **α)** ένα πίνακα από `struct student`, **β)** τη διεύθυνση ενός ακεραίου που αντιπροσωπεύει το πλήθος των συμπληρωμένων εγγραφών του πίνακα και **γ)** μία συμβολοσειρά. Η συνάρτηση επιστρέφει 1 αν γίνει έστω και μία διαγραφή, διαφορετικά 0.

Η συνάρτηση αναζητά όλες τις εγγραφές μέσα στον πίνακα όπου το όνομα ταυτίζεται με τη συμβολοσειρά που δίνεται ως τρίτη παράμετρος. Κάθε τέτοια εγγραφή που βρίσκει τη διαγράφει, αντιγράφοντας το τελευταίο στοιχείο του πίνακα στη θέση της εγγραφής που θέλουμε να διαγράψουμε και μειώνοντας το πλήθος εγγραφών του πίνακα κατά 1. Η διαδικασία επαναλαμβάνεται έως ότου διαγραφούν όλες οι εγγραφές που περιέχουν το όνομα που αναζητούμε.

**TIP:** Ξεκινήστε την αναζήτηση εγγραφών προς διαγραφή από το τέλος προς την αρχή. Με αυτό τον τρόπο εάν μία ή περισσότερες εγγραφές που βρίσκονται στο τέλος πίνακα πρέπει να διαγραφούν, αυτές διαγράφονται κατά προτεραιότητα. **Τι αποφεύγετε υιοθετώντας την παραπάνω τεχνική;**

**Χρησιμοποιήστε** την παραπάνω συνάρτηση στη `main`, στην επιλογή `delete` της `switch`, μετά την ανάγνωση του ονόματος προς διαγραφή που έχουμε ήδη υλοποιήσει για εσάς ώστε να διαγραφούν οι εμφανίσεις αυτού του ονόματος από τον πίνακα. Εάν γίνει έστω και μία διαγραφή εκτυπώνεται το μήνυμα "**Remaining students X**", ακολουθούμενο από χαρακτήρα αλλαγής γραμμής, όπου `X` είναι οι υπολοίπων μη κενές θέσεις του πίνακα. Διαφορετικά εκτυπώνει "**No student found!**" ακολουθούμενο από χαρακτήρα αλλαγής γραμμής.

**Βήμα 4ο:** Γράψτε μία συνάρτηση αναζήτησης η οποία λαμβάνει ως παραμέτρους **α)** ένα πίνακα από `struct student`, **β)** έναν ακεραίο που αντιπροσωπεύει το πλήθος των συμπληρωμένων εγγραφών του πίνακα και **γ)** μία συμβολοσειρά. Η συνάρτηση αναζητά την πρώτη εγγραφή μέσα στον πίνακα όπου το όνομα του φοιτητή ταυτίζεται με τη συμβολοσειρά που δίνεται ως τρίτη παράμετρος. Εάν βρεθεί εγγραφή επιστρέφει τη διεύθυνσή της, διαφορετικά επιστρέφει `NULL`.

**Χρησιμοποιήστε** την παραπάνω συνάρτηση στη `main`, στην επιλογή `search`, μετά την ανάγνωση του ονόματος προς αναζήτηση που έχουμε ήδη υλοποιήσει για εσάς, ώστε να βρεθεί η πρώτη αντίστοιχη εγγραφή. Εάν δεν βρεθεί εγγραφή εκτυπώνει το μήνυμα "**No student found!**" ακολουθούμενο από χαρακτήρα αλλαγής γραμμής, διαφορετικά εκτυπώνει την εγγραφή με τη βοήθεια της συνάρτησης του βήματος 2.

## Άσκηση 2 (προαιρετική στο lab, υποχρεωτική στο σπίτι)

Ξαναγράψτε το πρόγραμμα της άσκησης 1, αλλάζοντας όμως το πρόγραμμα ως εξής.

Δημιουργήστε ένα νέο struct το οποίο περιέχει δύο πεδία **α)** ένα πίνακα από struct της άσκησης 1 και **β)** έναν μη προσημασμένο ακέραιο που περιέχει το πλήθος των εγγραφών.

Προσαρμόστε τις συναρτήσεις της άσκησης 1 ώστε να αλλάξουν οι παράμετροι που λαμβάνουν αλλά όχι η λειτουργικότητα τους. **Κάντε τις απαραίτητες αλλαγές στη main ώστε να διατηρηθεί η ίδια λειτουργικότητα.**

**Βήμα 1ο:** Ξαναγράψτε τη συνάρτηση του βήματος 1 περνώντας ως μοναδική παράμετρο ένα δείκτη στο νέο struct.

**Βήμα 2ο:** Η συνάρτηση παραμένει ως έχει. Οι αλλαγές που θα κάνετε για αυτό το βήμα αφορούν τον κώδικα στη main.

**Βήμα 3ο:** Ξαναγράψτε τη συνάρτηση του βήματος 1 περνώντας δύο παραμέτρους **α)** ένα δείκτη στο νέο struct και **β)** μία συμβολοσειρά.

**Βήμα 4ο:** Ξαναγράψτε τη συνάρτηση του βήματος 1 περνώντας δύο παραμέτρους **α)** ένα δείκτη στο νέο struct και **β)** μία συμβολοσειρά.