

**ΣΕΤ ΑΣΚΗΣΕΩΝ 3****ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2019-2020****Προθεσμία: 22/12/2019, 23:59**

Διαβάστε ΟΛΗ την εκφώνηση προσεκτικά και “σχεδιάστε” το πρόγραμμά σας στο χαρτί. Αποφασίστε τι μεταβλητές θα χρειαστείτε, τι ονόματα θα τους δώσετε, αν χρειάζονται σταθερές κι αν ναι για ποιες ποσότητες, τι δομές ελέγχου θα χρησιμοποιήσετε για κάθε λειτουργία, αν και ποιες επιπλέον συναρτήσεις χρειάζονται. Κάθε φορά που ολοκληρώνετε ένα στάδιο, επιβεβαιώστε ότι το πρόγραμμά σας λειτουργεί σωστά, πριν προχωρήσετε παρακάτω.

Αν χρειάζεστε διευκρινίσεις ή έχετε προβλήματα, στείλτε μήνυμα στο χώρο συζητήσεων στο e-class. Προσοχή: δεν επιτρέπεται η δημοσίευση κώδικα στο eclass.

Η εργασία αυτή μπορεί να γίνει σε ομάδες μέχρι και 2 ατόμων. Δε χρειάζεται να είστε ομάδα με το ίδιο άτομο που είστε στο εργαστήριο ή που κάνατε την προηγούμενη εργασία. Μπορείτε να συζητάτε τις ασκήσεις με συμφοιτητές σας αλλά δεν επιτρέπεται η ανταλλαγή κώδικα με οποιονδήποτε τρόπο.

Οι ασκήσεις σας θα βαθμολογηθούν στα παρακάτω (χωρίς ιδιαίτερη σειρά):

- Ορθότητα υπολογισμών
- Σωστή χρήση δεικτών, struct, συναρτήσεων και πινάκων
- Αποτελεσματική χρήση κατάλληλων δομών, μεταβλητών, σταθερών κτλ.
- Γενική μορφοποίηση προγράμματος (στοίχιση, ονόματα μεταβλητών και σταθερών, κτλ.)
- Συμμόρφωση με τις προδιαγραφές
- Αποτελεσματικά σχόλια

Απαγορεύεται αυστηρά η χρήση goto, η χρήση gets και η χρήση καθολικών μεταβλητών.

Μπορείτε να υποθέσετε ότι η είσοδος από το πληκτρολόγιο θα δίνεται πάντα στη σωστή μορφή.

Στα μηνύματα εξόδου που τελειώνουν σε ':' υπάρχει ένας χαρακτήρας κενό (space) μετά το ':'.

**Οδηγίες υποβολής εργασίας στο Autolab**

1. Προσθέστε σε σχόλια στην αρχή του αρχείου `hw3.c` τα πλήρη ονόματα και ΑΕΜ των μελών της ομάδας. Παρακαλούμε να γράφετε τα σχόλια ΜΟΝΟ με αγγλικούς χαρακτήρες.
2. Κατασκευάστε ένα κατάλογο με όνομα `hw3submit` και αντιγράψτε μέσα σε αυτόν το `hw3.c`
3. Κατασκευάστε ένα αρχείο κειμένου με όνομα `team.txt` και προσθέστε σε αυτό τα πλήρη ονόματα και τα ΑΕΜ των μελών της ομάδας, ακόμη κι αν η ομάδα αποτελείται από ένα άτομο.
4. Κάντε δεξί κλικ στον κατάλογο `hw3submit` και επιλέξτε `Compress here as tar.gz`. Θα δημιουργηθεί ένα αρχείο με όνομα `hw3submit.tar.gz`.
5. Συνδεθείτε στο Autolab κι επιλέξτε το `hw3`.
  - Αν είστε ομάδα δύο ατόμων και δεν το έχετε ήδη κάνει, κατασκευάστε μια ομάδα μέσω της επιλογής `Group Options`.
  - Κάντε υποβολή το `hw3submit.tar.gz`.

## Δημιουργία προγράμματος αιθουσών

Σε αυτή την εργασία θα γράψετε ένα πρόγραμμα C το οποίο δημιουργεί το εβδομαδιαίο πρόγραμμα μαθημάτων για ένα έτος σπουδών σε ένα πανεπιστημιακό τμήμα. Το πρόγραμμα διατηρεί ένα πίνακα για τα μαθήματα και ένα πίνακα για τις αίθουσες και τοποθετεί τη διάλεξη ενός μαθήματος σε μία αίθουσα με βάση συγκεκριμένα κριτήρια που προσδιορίζονται παρακάτω.

Συνοπτικά, οι λειτουργίες που προδιαγράφει η συγκεκριμένη εργασία είναι οι εξής:

1. Εισαγωγή νέου μαθήματος
2. Εκτύπωση των μαθημάτων
3. Προγραμματισμός διάλεξης υφιστάμενου μαθήματος
4. Εκτύπωση του προγράμματος των αιθουσών
5. Διαγραφή μαθήματος

Οι διαλέξεις μπορούν να γίνονται τις εργάσιμες ημέρες της εβδομάδας (Δευτέρα έως και Παρασκευή) και ώρες μεταξύ 10.00π.μ. έως 3.00μ.μ. Η ελάχιστη διάρκεια μιας διάλεξης είναι 1 ώρα.

### Αρχεία

Σας παρέχονται τα αρχεία `libhw3.a` και `hw3.h` τα οποία δεν επιτρέπεται να αλλάξετε καθώς και το αρχείο `hw3.c` στο οποίο θα προσθέσετε τον κώδικά σας. Το αρχείο `libhw3.a` περιέχει την υλοποίηση της συνάρτησης `print_menu` που εκτυπώνει το μενού του προγράμματος. Το μενού διαθέτει τις εξής επιλογές:

- a: (add course) προσθήκη μαθήματος
- c: (print courses) εκτύπωση αποθηκευμένων μαθημάτων
- s: (schedule lecture) προγραμματισμός διάλεξης για ένα μάθημα
- p: (print schedule) εκτύπωση του προγράμματος
- r: (remove course) διαγραφή μαθήματος
- q: (quit) τερματισμός του προγράμματος

Για να μεταγλωττίσετε το πρόγραμμά σας, βεβαιωθείτε ότι το αρχείο `libhw3.a` βρίσκεται στον ίδιο κατάλογο με τα αρχεία `hw3.c`, `hw3.h` και γράψτε:

```
gcc -Wall -g hw3.c -o hw3 -lhw3 -L.
```

### Προσδιορισμός των δομών δεδομένων του προγράμματος

Ορίστε ένα struct που περιγράφει ένα μάθημα με τα εξής πεδία:

- α) αναγνωριστικό μαθήματος (ακέραιος),
- β) όνομα καθηγητή (πίνακας χαρακτήρων μεγέθους `MAX_NAME_LEN`) και
- γ) αριθμός φοιτητών που είναι εγγεγραμμένοι στο μάθημα (ακέραιος).

Το πρόγραμμα δημιουργεί ένα πίνακα μαθημάτων μεγέθους `MAX_COURSES`, όπου τα περιεχόμενα κάθε μαθήματος είναι όλα αρχικοποιημένα σε μηδέν.

Ορίστε ένα struct που περιγράφει μια αίθουσα διδασκαλίας με τα εξής πεδία:

- α) χωρητικότητα αίθουσας (ακέραιος)
- β) εβδομαδιαίο πρόγραμμα αίθουσας (διδιάστατος πίνακας δεικτών σε μάθημα, με μέγεθος ίσο με τον αριθμό των ωρών που μπορεί να είναι ανοιχτή μία αίθουσα επί των αριθμό των εργάσιμων ημερών της εβδομάδας (`WORKING_HOURS` x `WORKING_DAYS`)).

Το πρόγραμμα δημιουργεί ένα πίνακα αιθουσών μεγέθους `MAX_ROOMS`, όπου η χωρητικότητα κάθε αίθουσας είναι αρχικοποιημένη σε μηδέν και ο πίνακας δεικτών για το πρόγραμμα της αίθουσας έχει

## Στάδιο 0 - Διάβασμα της πληροφορίας των αιθουσών από τη γραμμή εντολών

Το πρόγραμμα που θα γράψετε δέχεται ως ορίσματα μια σειρά ακεραίων που αντιπροσωπεύουν την χωρητικότητα της κάθε αίθουσας σε μη φθίνουσα σειρά. Για παράδειγμα, για τρεις αίθουσες με χωρητικότητες 15, 40 και 15 το πρόγραμμα θα πρέπει να κληθεί ως εξής: `./hw3 15 15 40`

Στην περίπτωση που ο χρήστης δώσει ως ορίσματα (α) έστω και μία μη θετική ακεραία τιμή, ή (β) τη χωρητικότητα των αιθουσών σε φθίνουσα σειρά, ή (γ) περισσότερες τιμές από το μέγιστο αριθμό αιθουσών (MAX\_ROOMS) εκτυπώνεται το μήνυμα `"Incorrect command-line arguments!"` ακολουθούμενο από χαρακτήρα αλλαγής γραμμής και το πρόγραμμα τερματίζει.

Εφόσον δεν υπάρχει πρόβλημα με τα ορίσματα, θα πρέπει να αρχικοποιείται κατάλληλα ο πίνακας αιθουσών με βάση τις χωρητικότητες που δόθηκαν μέσω των ορισμάτων. **Προσοχή:** Η αποθήκευση πρέπει να γίνεται με τη σειρά που δόθηκαν οι αντίστοιχες χωρητικότητες (μη φθίνουσα σειρά).

Στην συνέχεια, καλείται σε επανάληψη η συνάρτηση `print_menu` και διαβάζεται από την είσοδο του προγράμματος η επιλογή του χρήστη. Σε αυτή τη φάση, το πρόγραμμα πρέπει να χειρίζεται μόνο την επιλογή `a` στην οποία περίπτωση το πρόγραμμα απλά τερματίζει. Κάθε άλλη επιλογή δεν έχει κάποιο αποτέλεσμα, και απλά επαναλαμβάνεται η εκτύπωση του μενού και η ανάγνωση της επόμενης επιλογής του χρήστη.

## Στάδιο 1 - Προσθήκη νέου μαθήματος

Γράψτε μία συνάρτηση η οποία λαμβάνει ως παραμέτρους ένα πίνακα μαθημάτων και το αναγνωριστικό ενός μαθήματος και επιστρέφει ένα δείκτη σε μάθημα. Η συνάρτηση αναζητεί το μάθημα με το συγκεκριμένο αναγνωριστικό μέσα στο πίνακα μαθημάτων και αν το βρει επιστρέφει τη διεύθυνση του, αν δεν το βρει επιστρέφει τη διεύθυνση της πρώτης κενής θέσης του πίνακα, ενώ αν δεν το βρει και δεν υπάρχει κενή θέση στον πίνακα επιστρέφει NULL.

Γράψτε μία δεύτερη συνάρτηση η οποία λαμβάνει ως παραμέτρους ένα πίνακα μαθημάτων, το αναγνωριστικό ενός μαθήματος, το πλήθος φοιτητών σε αυτό το μάθημα και το όνομα του διδάσκοντος καθηγητή κι επιστρέφει ακεραίο. Η συνάρτηση, με τη βοήθεια της προηγούμενης συνάρτησης, αναζητεί το συγκεκριμένο μάθημα στον πίνακα των μαθημάτων. Εάν το εντοπίσει επιστρέφει 0. Εάν δεν το εντοπίσει, το προσθέτει στην πρώτη κενή θέση του πίνακα μαθημάτων κι επιστρέφει 1. Εάν δεν το εντοπίσει και δεν υπάρχουν κενές θέσεις στον πίνακα μαθημάτων, επιστρέφει -1. Λάβετε υπόψη ότι σε οποιαδήποτε χρονική στιγμή τα μαθήματα δεν είναι απαραίτητα αποθηκευμένα σε διαδοχικές θέσεις του πίνακα.

Συμπληρώστε κώδικα στη συνάρτηση `main`, ώστε όταν ο χρήστης επιλέξει τη λειτουργία προσθήκης μαθήματος (επιλογή `a`) το πρόγραμμα εκτυπώνει το μήνυμα `"professor course students: "` και διαβάζει το όνομα του διδάσκοντος καθηγητή (το οποίο δεν περιέχει κενά), το αναγνωριστικό του μαθήματος και τον αριθμό των φοιτητών που είναι εγγεγραμμένοι στο συγκεκριμένο μάθημα. Υποθέστε ότι ως αναγνωριστικό και πλήθος φοιτητών θα δίνονται πάντα θετικοί αριθμοί (δε χρειάζεται έλεγχος για το αν η ανάγνωση πέτυχε). Μετά, καλεί τη δεύτερη εκ των δύο παραπάνω συναρτήσεων προκειμένου να προστεθεί ένα νέο μάθημα στον πίνακα μαθημάτων. Αναλόγως με την τιμή επιστροφής της συνάρτησης, το πρόγραμμα εκτυπώνει το μήνυμα `"C exists"` ή το μήνυμα `"C added"` ή το μήνυμα `"No space"`, όπου `C` το αναγνωριστικό του μαθήματος, ακολουθούμενο από χαρακτήρα αλλαγής γραμμής.

## Στάδιο 2 - Εκτύπωση των αποθηκευμένων μαθημάτων

Γράψτε μία συνάρτηση η οποία λαμβάνει ως παράμετρο ένα πίνακα μαθημάτων και επιστρέφει `void`. Η συνάρτηση εκτυπώνει στην καθιερωμένη έξοδο όλα τα μαθήματα που είναι αποθηκευμένα στον πίνακα. Για κάθε μάθημα εκτυπώνει μήνυμα της μορφής `"[C] P S"` ακολουθούμενο από χαρακτήρα αλλαγής

γραμμής, όπου `C` είναι το μοναδικό αναγνωριστικό του μαθήματος, `P` είναι το όνομα του καθηγητή και `S` ο αριθμός των εγγεγραμμένων φοιτητών στο μάθημα.

Συμπληρώστε κώδικα στη συνάρτηση `main`, ώστε όταν ο χρήστης επιλέξει τη λειτουργία εκτύπωσης των μαθημάτων (επιλογή `c`) να καλείται η παραπάνω συνάρτηση με όρισμα τον πίνακα των μαθημάτων.

### Στάδιο 3 - Προγραμματισμός διάλεξης για ένα μάθημα

Γράψτε μία συνάρτηση η οποία λαμβάνει ως παραμέτρους τον πίνακα αιθουσών, δείκτη στο μάθημα για το οποίο επιθυμούμε να προγραμματιστεί διάλεξη και την επιθυμητή διάρκεια της διάλεξης, κι επιστρέφει ακέραιο. Η συνάρτηση επιχειρεί να τοποθετήσει τη νέα διάλεξη του μαθήματος στον πίνακα αιθουσών.

Εξετάζει μία-μία τις αίθουσες μέχρι να βρει την πρώτη που χωράει τον αριθμό των φοιτητών του μαθήματος. Στην αίθουσα προς εξέταση επιχειρεί να βρει την πρώτη ελεύθερη θέση στο πρόγραμμα μαθημάτων αυτής ελέγχοντας α) τη διαθεσιμότητα της αίθουσας ανά ώρα (αρχίζοντας από 10π.μ.) και κατόπιν ανά ημέρα (αρχίζοντας από Δευτέρα) και β) αν ο καθηγητής του μαθήματος δεν έχει μάθημα την ίδια ώρα σε κάποια άλλη αίθουσα.

Αν βρεθεί κατάλληλη μέρα/ώρα σε αυτή την αίθουσα, η διάλεξη καταχωρείται στην αντίστοιχη θέση και η συνάρτηση επιστρέφει τη θέση της αίθουσας στον πίνακα αιθουσών. Αν δε βρεθεί κατάλληλη θέση σε αυτή την αίθουσα, η αναζήτηση επαναλαμβάνεται για την επόμενη σε χωρητικότητα αίθουσα. Η συνάρτηση επιστρέφει `-1` αν δε βρεθεί κατάλληλη μέρα/ώρα σε καμία αίθουσα.

Συμπληρώστε κώδικα στη συνάρτηση `main` ώστε όταν ο χρήστης επιλέξει τη λειτουργία προγραμματισμού διάλεξης (επιλογή `s`) για ένα μάθημα, το πρόγραμμα εκτυπώνει το μήνυμα `"course duration: "` και διαβάζει το αναγνωριστικό του μαθήματος και τη διάρκεια της διάλεξης. Αν η διάρκεια της διάλεξης είναι μικρότερη του 1, εκτυπώνει το μήνυμα `"Invalid duration"` ακολουθούμενο από χαρακτήρα αλλαγής γραμμής. Αν το μάθημα δεν υφίσταται, εκτυπώνει το μήνυμα `"C not found"` ακολουθούμενο από χαρακτήρα αλλαγής γραμμής, όπου `C` το αναγνωριστικό του μαθήματος. Διαφορετικά, καλεί την παραπάνω συνάρτηση για να προγραμματίσει τη διάλεξη και σε περίπτωση επιτυχίας εκτυπώνει `"C scheduled in R"` όπου `C` το αναγνωριστικό του μαθήματος και `R` η θέση (στον πίνακα αιθουσών) της αίθουσας στην οποία καταχωρήθηκε η διάλεξη γι αυτό το μάθημα, ενώ σε περίπτωση αποτυχίας εκτυπώνει `"C not scheduled"` όπου `C` το αναγνωριστικό του μαθήματος. Στη συνέχεια εκτυπώνει χαρακτήρα αλλαγής γραμμής.

### Στάδιο 4 - Εκτύπωση του προγράμματος των αιθουσών

Γράψτε μία συνάρτηση η οποία λαμβάνει ως παράμετρο ένα δείκτη σε μία αίθουσα, έχει επιστρεφόμενο τύπο `void` και εκτυπώνει το εβδομαδιαίο πρόγραμμα της αίθουσας ως εξής:

Εκτυπώνει χαρακτήρα αλλαγής γραμμής και τη συμβολοσειρά `"Capacity: X"` ακολουθούμενη από χαρακτήρα αλλαγής γραμμής, όπου `X` ο ακέραιος που προσδιορίζει τη χωρητικότητα της αίθουσας.

Στη συνέχεια, εκτυπώνει επτά κενούς χαρακτήρες και όσες ημέρες από τη συμβολοσειρά `"MON TUE WED THU FRI"` αντιστοιχούν στο `WORKING_DAYS` ακολουθούμενη από χαρακτήρα αλλαγής γραμμής.

Για κάθε ώρα της ημέρας εκτυπώνει χαρακτήρα αλλαγής γραμμής, τη συμβολοσειρά `"DD:00"` όπου `DD` η ακέραια ώρα της ημέρας με εύρος δύο ψηφίων, και δύο κενά. Οι ώρες μετά τις 12:00 το μεσημέρι εκτυπώνονται ως 13:00, 14:00 κ.ο.κ. Για κάθε ημέρα της εβδομάδας, εάν υπάρχει διάλεξη τη συγκεκριμένη ώρα, εκτυπώνει το αναγνωριστικό του μαθήματος με εύρος 4 χαρακτήρες και στοιχισμένο αριστερά, διαφορετικά κενό, παύλα και δύο κενά.

Στο τέλος του εβδομαδιαίου προγράμματος εκτυπώνει χαρακτήρα αλλαγής γραμμής.

Γράψτε μία δεύτερη συνάρτηση η οποία λαμβάνει ως παράμετρο τον πίνακα αιθουσών, έχει επιστρεφόμενο τύπο `void` κι εκτυπώνει το εβδομαδιαίο πρόγραμμα όλων των αιθουσών

χρησιμοποιώντας την παραπάνω συνάρτηση, με τη σειρά που είναι αποθηκευμένες οι αίθουσες στον πίνακα αιθουσών.

Συμπληρώστε κώδικα στη συνάρτηση `main` ώστε όταν ο χρήστης επιλέξει τη λειτουργία εκτύπωσης του προγράμματος των αιθουσών (επιλογή `p`) να καλείται η δεύτερη εκ των παραπάνω συναρτήσεων.

## Στάδιο 5 - Διαγραφή μαθήματος

Γράψτε μία συνάρτηση η οποία λαμβάνει ως παραμέτρους τον πίνακα μαθημάτων, το αναγνωριστικό ενός μαθήματος και τον πίνακα αιθουσών κι επιστρέφει ακέραιο. Η συνάρτηση αναζητεί το μάθημα με το δοθέν αναγνωριστικό στον πίνακα των μαθημάτων. Εφόσον το μάθημα υπάρχει, αφαιρεί όλες τις διαλέξεις του μαθήματος που είναι καταχωρημένες στον πίνακα των αιθουσών, αποθηκεύοντας στις αντίστοιχες θέσεις του πίνακα δεικτών κάθε αίθουσας την τιμή `NULL`. Στη συνέχεια διαγράφει το μάθημα από τον πίνακα των μαθημάτων. Ο τρόπος διαγραφής είναι δική σας επιλογή, αρκεί να μη μετακινηθούν τα περιεχόμενα των κελιών του πίνακα που αντιστοιχούν σε ήδη υπάρχοντα μαθήματα. Η συνάρτηση επιστρέφει μηδέν σε περίπτωση αποτυχίας (αν το εν λόγω μάθημα δεν βρέθηκε) ή έναν ακέραιο διαφορετικό του μηδενός σε περίπτωση επιτυχίας.

Συμπληρώστε τη συνάρτηση `main` ώστε όταν ο χρήστης επιλέξει τη λειτουργία διαγραφής μαθήματος (επιλογή `r`) εκτυπώνει το μήνυμα `"course: "` (υπάρχει κενό μετά τον χαρακτήρα `:`), διαβάζει το αναγνωριστικό του μαθήματος και καλεί την παραπάνω συνάρτηση προκειμένου να διαγράψει το μάθημα. Σε περίπτωση που το μάθημα διαγραφεί με επιτυχία εκτυπώνει το μήνυμα `"C deleted"`, διαφορετικά εκτυπώνει το μήνυμα `"C not deleted"` ακολουθούμενο από χαρακτήρα αλλαγής γραμμής, όπου `C` το αναγνωριστικό του μαθήματος.