

ΣΕΤ ΑΣΚΗΣΕΩΝ 4

ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2017-2018

Προθεσμία: Τετάρτη 10/1/2018, 22:00

Διαβάστε πριν ξεκινήσετε

Διαβάστε την εκφώνηση προσεκτικά. Ολοκληρώστε κάθε ένα στάδιο πριν προχωρήσετε στο επόμενο, κι αποθηκεύστε σε ξεχωριστά αρχεία ενδιάμεσες σωστές μορφές του προγράμματός σας ώστε να μπορείτε να επανέλθετε σε αυτές αν κάνετε κάποιο λάθος. Το μέρος Β χτίζει πάνω στο μέρος Α, επομένως βεβαιωθείτε ότι το Α είναι σωστό πριν προχωρήσετε.

Μη διστάζετε να ζητήσετε βοήθεια! Χρησιμοποιήστε κατά προτίμηση το χώρο συζητήσεων στο eclass ή, αν είναι απαραίτητο, email (π.χ. αν πραγματικά επιβάλλεται να στείλετε κάποιο κομμάτι κώδικα μαζί με το μήνυμά σας).

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

Ξεκινήστε νωρίς! Ο προγραμματισμός είναι πάντα ΠΟΛΥ πιο χρονοβόρος από ότι περιμένετε.

Εκπρόθεσμες ασκήσεις δε γίνονται δεκτές.

Η άσκησή σας θα βαθμολογηθεί στα παρακάτω (χωρίς ιδιαίτερη σειρά):

- Ορθότητα
- Γενική μορφοποίηση προγράμματος (στοίχιση, ονόματα μεταβλητών/συναρτήσεων/σταθερών κτλ.)
- Σχεδιασμός προγράμματος και αποτελεσματική χρήση κατάλληλων δομών, μεταβλητών, σταθερών, struct, enum κτλ.
- Σωστή δημιουργία και χρήση συναρτήσεων
 - Κάθε συνάρτηση πρέπει να είναι καλά ορισμένη: να κάνει μια σαφώς ορισμένη δουλειά, να μην έχει περιττές παραμέτρους, να είναι σχετικά μικρή.
 - Αποφύγετε την άσκοπη επανάληψη κώδικα. Αν δείτε ότι κάνετε copy+paste κομμάτια κώδικα, τότε μάλλον χρειάζεστε συνάρτηση για τη συγκεκριμένη λειτουργία.
- Συμμόρφωση με τις προδιαγραφές
- Σωστή δέσμευση και αποδέσμευση δυναμικής μνήμης και χρήση λιστών. Δώστε ιδιαίτερη προσοχή στο πότε πρέπει να αποδεσμεύετε μνήμη.
- Παραμετροποίηση του προγράμματος. Το πρόγραμμα θα πρέπει να λειτουργεί σωστά ακόμη κι αν αλλάξουν οι τιμές των σταθερών που έχουν οριστεί με #define και θα πρέπει να είναι εύκολο να γίνουν αλλαγές σε πεδία enum.
- Αποτελεσματικά σχόλια, σύμφωνα με τους κανόνες σχολιασμού φυλλαδίου με τίτλο "Πρότυπα σχολιασμού προγραμμάτων".

Προσοχή: Απαγορεύεται να χρησιμοποιήσετε καθολικές/static μεταβλητές, goto, gets και ότι άλλο έχουμε "απαγορεύσει" στις εργασίες.

Μέρος Α: Ένα ραντεβού τη μέρα.

Εισαγωγή

Στο πρώτο μέρος της εργασίας θα υλοποιήσετε ένα ημερολόγιο που δίνει τη δυνατότητα να καταγραφεί το πολύ ένα ραντεβού σε κάθε ημέρα που καλύπτει. Το πρόγραμμά σας θα πρέπει να δίνει τη δυνατότητα στο χρήστη να προσθέσει ένα νέο ραντεβού στο ημερολόγιο, να αφαιρέσει όλα τα ραντεβού που ικανοποιούν μια συγκεκριμένη συνθήκη και να εκτυπώσει όλα τα ραντεβού.

Ακολουθούν λεπτομερείς οδηγίες για το πώς πρέπει να λειτουργεί το πρόγραμμά σας και στάδια κατασκευής του. ΜΗΝ προσπαθήσετε να γράψετε όλο το πρόγραμμα σε ένα βήμα γιατί θα κάνετε λάθη και θα σας πάρει πολύ περισσότερο χρόνο.

Στάδιο 0: Μελέτη του αρχείου hw4partA.c

Ανοίξτε το αρχείο hw4partA.c και μελετήστε τον κώδικα που περιέχει.

Παρατηρήστε πως έχουμε χρησιμοποιήσει ένα `enum` για να δώσουμε ονόματα στις λειτουργίες που θέλουμε να υλοποιήσουμε. Αυτό κάνει το πρόγραμμά μας πιο κατανοητό, και μας διευκολύνει αν θέλουμε να κάνουμε κάποια αλλαγή: θα είναι εύκολο να προσθέσουμε ή να αφαιρέσουμε μια λειτουργία και να κάνουμε τις αντίστοιχες αλλαγές στις συναρτήσεις που επηρεάζονται, όπως για παράδειγμα η `printMenu` που βλέπετε στο αρχείο.

Επίσης προσέξτε τη μεταβλητή `debug`. Είναι η μόνη καθολική μεταβλητή που επιτρέπεται να χρησιμοποιηθεί σε αυτό το homework. Σκοπός της είναι να καταγράφει αν θέλουμε να εκτυπωθούν περισσότερες πληροφορίες για την εκτέλεση του προγράμματος. Αρχικά έχει την τιμή `FALSE` το οποίο σημαίνει ότι δε θέλουμε επιπλέον πληροφορίες, αλλά αυτό μπορεί να αλλάξει ανάλογα με την είσοδο που μπορεί να δώσει ο χρήστης από τη γραμμή εντολών.

Στις ενότητες που ακολουθούν, θα εμφανίζουμε με **μπλε χρώμα** τα μηνύματα που εκτυπώνονται πάντα και με **κόκκινο χρώμα** τα μηνύματα που εκτυπώνονται μόνο όταν η `debug` έχει την τιμή `TRUE`.

Στάδιο 1: Διαχείριση ορισμάτων προγράμματος, δημιουργία και αρχικοποίηση δομών (*)

Το πρόγραμμα μπορεί να εκτελεστεί είτε χωρίς ορίσματα είτε με τα ορίσματα που περιγράφονται παρακάτω. Υποθέτουμε ότι το εκτελέσιμο πρόγραμμα λέγεται hw4.

```
./hw4 -debug on
```

Σε αυτή την περίπτωση πρέπει η μεταβλητή `debug` να παίρνει την τιμή `TRUE`

```
./hw4 -debug off
```

Σε αυτή την περίπτωση πρέπει η μεταβλητή `debug` να παίρνει την τιμή `FALSE`

Γράψτε κώδικα στη `main` ώστε να ανιχνεύετε αν έχουν δοθεί ορίσματα κι αν ναι, να ελέγχετε αν είναι κάτι από τα παραπάνω και να θέτετε αντίστοιχα τη μεταβλητή `debug`. Σε κάθε άλλη περίπτωση, η μεταβλητή `debug` παραμένει `FALSE`. Αν έχουν δοθεί λάθος ορίσματα, το πρόγραμμα εκτυπώνει το μήνυμα

Ignoring incorrect arguments.

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής** και συνεχίζει κανονικά την εκτέλεση. Γράψτε το πρόγραμμά σας ώστε να είναι εύκολο να ελεγχθούν επιπλέον ορίσματα αν χρειαστεί (με άλλα λόγια, μην ελέγχετε απλά αν για παράδειγμα το δεύτερο όρισμα αρχίζει από -, αλλά ελέγξτε αν είναι ακριβώς η λέξη `-debug`

Για κάθε ραντεβού θέλουμε να καταγράψουμε την ώρα που ξεκινά, την ώρα που τελειώνει, μια σύντομη περιγραφή, και τι κατηγορία ραντεβού είναι.

Η ώρα εκκίνησης ή τερματισμού ενός ραντεβού θα αναπαρασταθεί από ένα `struct` με δύο πεδία:

- Ένα ακέραιο που αναπαριστά την ώρα
- Ένα ακέραιο που αναπαριστά το λεπτό

Η κατηγορία ενός ραντεβού θα αναπαρασταθεί από ένα enum το οποίο παίρνει τις τιμές MEETING, CLASS, SKYPECALL.

Επομένως, το ραντεβού θα αναπαρασταθεί από ένα struct το οποίο περιέχει τα εξής πεδία:

- Περιγραφή: Δείκτης προς χαρακτήρα (συμβολοσειρά)
- Κατηγορία: τύπος enum
- Έναρξη: τύπος struct που αναπαριστά ώρα
- Λήξη: τύπος struct που αναπαριστά ώρα

Το ημερολόγιο θα αναπαρασταθεί ως πίνακας μεγέθους όσες είναι οι μέρες που θέλουμε να καλύπτει. Κάθε κελί του πίνακα θα περιέχει ένα δείκτη προς ένα ραντεβού. Αρχικά, δεν υπάρχουν ραντεβού στο ημερολόγιο.

Προσθέστε κώδικα στο `hw4partA.c` που να δημιουργεί όλα τα struct και enum που χρειάζονται για να αναπαρασταθεί ένα ραντεβού.

Προσθέστε κώδικα στη `main` που να δημιουργεί ένα πίνακα από δείκτες σε ραντεβού, μεγέθους DAYS. Από εδώ και πέρα θα τον αποκαλούμε "ημερολόγιο".

Γράψτε μια συνάρτηση η οποία παίρνει ως παράμετρο το ημερολόγιο και το αρχικοποιεί ώστε να μην περιέχει ραντεβού.

Προσθέστε κώδικα στη `main` που να καλεί την παραπάνω συνάρτηση.

Ολοκληρώστε αυτό το στάδιο και βεβαιωθείτε ότι λειτουργεί σωστά πριν προχωρήσετε στο επόμενο.

Στάδιο 2: Στοιχειώδεις λειτουργίες (*)

Σε αυτό το στάδιο θα υλοποιήσετε τη βασική αλληλεπίδραση του προγράμματος με το χρήστη. Θα αναπτύξετε το πρόγραμμά σας χρησιμοποιώντας τα λεγόμενα stubs: συναρτήσεις που δεν περιέχουν κώδικα ή περιέχουν κάτι πολύ στοιχειώδες. Στόχος είναι να βεβαιωθείτε ότι η βασική δομή του προγράμματός σας λειτουργεί σωστά. Αργότερα θα προσθέτετε κώδικα σε μια συνάρτηση τη φορά, και θα προχωράτε μόνο αφότου έχετε βεβαιωθεί ότι λειτουργεί σωστά το πρόγραμμα μέχρι στιγμής. Έτσι θα είναι πολύ εύκολο να απομονώσετε λάθη (αν το πρόγραμμα δούλευε πριν υλοποιηθεί η συνάρτηση A και χάλασε μετά την υλοποίηση της A, τότε το πρόβλημα είναι μάλλον στην A).

Γράψτε 4 συναρτήσεις (stubs) κάθε μία από τις οποίες παίρνει ως παράμετρο το ημερολόγιο. Σκοπός της κάθε συνάρτησης είναι να υλοποιεί μία λειτουργία του ημερολογίου (προσθήκη ραντεβού, σβήσιμο με βάση κατηγορία, σβήσιμο με βάση περιγραφή, εκτύπωση). Σε κάθε συνάρτηση απλά προσθέστε μια printf που εκτυπώνει ένα σύντομο μήνυμα για το τι θα κάνει η συνάρτηση όταν υλοποιηθεί πλήρως.

Γράψτε μια συνάρτηση (stub) η οποία παίρνει ως παράμετρο το ημερολόγιο. Σκοπός της θα είναι να απελευθερώνει ό,τι δυναμική μνήμη έχει δεσμευθεί για το ημερολόγιο. Σε αυτό το στάδιο, απλά προσθέστε μια printf που εκτυπώνει ένα σύντομο μήνυμα για το τι θα κάνει η συνάρτηση όταν υλοποιηθεί πλήρως.

Γράψτε μια συνάρτηση η οποία παίρνει ως παράμετρο το ημερολόγιο και κάνει τα παρακάτω:

- Εκτυπώνει το μενού λειτουργιών.
- Διαβάζει από το χρήστη την επιλογή του (ακέραιος).
- Ανάλογα με την επιλογή, καλεί μια από τις 5 συναρτήσεις που γράψατε πριν.
- Αν δε δοθεί έγκυρη επιλογή, εκτυπώνει το μήνυμα
`ERROR: Incorrect selection. Try again.`
ακολουθούμενο από χαρακτήρα αλλαγής γραμμής.
- Εφόσον ο χρήστης δεν έχει επιλέξει EXIT, επαναλαμβάνονται όλα τα παραπάνω βήματα.

Ολοκληρώστε αυτό το στάδιο και βεβαιωθείτε ότι λειτουργεί σωστά πριν προχωρήσετε στο επόμενο.

Στάδιο 3: Υλοποίηση δημιουργίας και εισαγωγής νέου ραντεβού ()**

Σε αυτό το στάδιο θα προσθέσετε κώδικα στο stub που ορίσατε για τη δημιουργία ενός νέου ραντεβού και την εισαγωγή του στο ημερολόγιο. Διαβάστε προσεκτικά την περιγραφή όλου του σταδίου πριν ξεκινήσετε. Επειδή η συνάρτηση είναι μεγάλη, σκεφτείτε τι επιμέρους λειτουργίες μπορείτε να κάνετε με βοηθητικές συναρτήσεις ώστε να είναι πιο συνοπτικός ο κώδικας που θα εμφανίζεται μέσα στη βασική σας συνάρτηση.

Τελικά, η συνάρτησή σας πρέπει να κάνει τα παρακάτω:

Ζητά από το χρήστη να προσδιορίσει την κατηγορία του ραντεβού του εκτυπώνοντας το μενού:

```
Select category:  
0. MEETING  
1. CLASS  
2. SKYPE CALL  
-->
```

Πριν από τις ακέραιες τιμές υπάρχει χαρακτήρας `tab`.

Το πρόγραμμα διαβάζει την κατάλληλη τιμή (hint: την εκτύπωση του μενού και την ανάγνωση κατηγορίας μπορείτε να τα κάνετε σε μία βοηθητική συνάρτηση). Αν δε δοθεί έγκυρη επιλογή, εκτυπώνει το μήνυμα

ERROR: Incorrect selection. Try again.

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής** κι επαναλαμβάνει την εκτύπωση του μενού και ανάγνωση.

Ζητά από το χρήστη να προσδιορίσει την περιγραφή του ραντεβού του εκτυπώνοντας το μήνυμα:

```
Enter short description:
```

ακολουθούμενο από ένα χαρακτήρα κενό (space). Διαβάζει την περιγραφή, η οποία είναι μια λέξη χωρίς κενά. Αν έχει ενεργοποιηθεί η debug, τότε εκτυπώνει το μήνυμα:

You entered "D".

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής**, όπου D είναι η περιγραφή που διαβάστηκε.

Ζητά από το χρήστη να προσδιορίσει την ώρα έναρξης του ραντεβού του εκτυπώνοντας το μήνυμα:

```
Enter start time in the form HH:MM :
```

ακολουθούμενο από **χαρακτήρα κενό** (space). Διαβάζει την ώρα που θα δώσει ο χρήστης (η οποία θα πρέπει να δοθεί στη μορφή HH:MM, για παράδειγμα 8:30). Επιβεβαιώνει ότι η ώρα που δόθηκε είναι εντός των ορίων που επιτρέπεται να κανονιστεί ραντεβού (δείτε #define στον κώδικα). Αν δεν είναι, εκτυπώνει το μήνυμα

ERROR: Invalid hour or minute. Try again.

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής**, κι επαναλαμβάνει αυτό το επιμέρους βήμα.

Ζητά από το χρήστη να προσδιορίσει τη διάρκεια του ραντεβού του εκτυπώνοντας το μήνυμα:

```
Enter duration in minutes :
```

ακολουθούμενο από **χαρακτήρα κενό** (space). Εάν δοθεί αρνητική τιμή, εκτυπώνει το μήνυμα

ERROR: Invalid duration. Try again.

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής**, κι επαναλαμβάνει αυτό το επιμέρους βήμα.

Αν έχει δοθεί έγκυρη τιμή, υπολογίζει την ώρα λήξης του ραντεβού με βάση την ώρα έναρξης και τη διάρκεια. Αν η ώρα λήξης είναι εκτός των ορίων που επιτρέπεται να κανονιστεί ραντεβού εκτυπώνει το μήνυμα

ERROR: Appointment exceeds end of workday. Try again.

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής**, κι επαναλαμβάνει αυτό το επιμέρους βήμα.

Εφόσον έχει δοθεί σωστή διάρκεια, κι αν το debug είναι ενεργοποιημένο, το πρόγραμμα εκτυπώνει το μήνυμα:

You entered hh:mm - HH:MM.

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής**, όπου hh η ώρα έναρξης, mm το λεπτό έναρξης, HH η ώρα λήξης, MM το λεπτό λήξης. Τόσο οι ώρες όσο και τα λεπτά πρέπει να καταλαμβάνουν από 2 θέσεις και αν είναι μονοψήφια να ξεκινούν από 0 (για παράδειγμα, εκτυπώνετε 08:30 και όχι 8:30).

Ζητά από το χρήστη να προσδιορίσει τη μέρα του ραντεβού του και τότε επαναλαμβάνεται, εκτυπώνοντας το μήνυμα:

`Enter day (1 - D) and repeat interval (>= 0):`

ακολουθούμενο από **χαρακτήρα κενό** (space), όπου D το πλήθος ημερών στο ημερολόγιο. Το πρόγραμμα διαβάζει την πρώτη ημέρα στην οποία θα μπει το ραντεβού και κάθε πόσες ημέρες επαναλαμβάνεται. Αν η μέρα είναι εκτός των ορίων [1-D] ή το διάστημα επανάληψης αρνητικό, το πρόγραμμα εκτυπώνει το μήνυμα:

`ERROR: Invalid day or interval. Try again.`

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής**, κι επαναλαμβάνει αυτό το επιμέρους βήμα.

Τώρα είστε έτοιμοι να προσθέσετε το ραντεβού στο ημερολόγιο:

Κατασκευάστε ένα νέο ραντεβού και δώστε του τις τιμές που διαβάσατε από το χρήστη. Εισάγετε το ραντεβού στη μέρα που έδωσε ο χρήστης, εφόσον δεν υπάρχει ήδη κάποιο ραντεβού εκεί. Προσοχή: ο χρήστης "μετρά" τις μέρες ξεκινώντας από το 1, αλλά στον πίνακα ξεκινάτε από το 0. Αν έχει δοθεί μη-μηδενικό διάστημα επανάληψης, κατασκευάστε όσα αντίγραφα του ραντεβού χρειάζεται και προσθέστε τα στις αντίστοιχες μέρες (χωρίς να υπερβείτε το μέγιστο πλήθος ημερών). Αν υπάρχει ήδη ραντεβού σε μια μέρα, δεν προσθέτετε νέο.

Για παράδειγμα, αν υποθέσουμε ότι το DAYS είναι 30, η πρώτη μέρα είναι 4 και το διάστημα επανάληψης 10, τότε θα πρέπει να προστεθούν αντίγραφα του ραντεβού στις μέρες 4, 14, 24.

Αν το debug είναι ενεργοποιημένο, κάθε φορά που προσθέτετε το ραντεβού εκτυπώστε το μήνυμα:

`Added appointment on day D.`

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής**, όπου D η μέρα στην οποία προστέθηκε το ραντεβού (μετρώντας από το 1).

Ολοκληρώστε αυτό το στάδιο και βεβαιωθείτε ότι λειτουργεί σωστά πριν προχωρήσετε στο επόμενο.

Στάδιο 4: Υλοποίηση εκτύπωσης ημερολογίου (*)

Σε αυτό το στάδιο θα προσθέσετε κώδικα στο stub που ορίσατε για την εκτύπωση του ημερολογίου.

Για κάθε ημέρα του ημερολογίου η οποία περιέχει ένα ραντεβού, εκτυπώστε:

```
=====
DAY   X
      CAT (descr), hh:mm - HH:MM
```

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής**. X είναι ο αύξων αριθμός της ημέρας (ξεκινώντας το μέτρημα από 1). CAT είναι η κατηγορία κι εμφανίζεται μετά από χαρακτήρα tab. descr είναι η περιγραφή. hh:mm είναι η έναρξη και HH:MM η λήξη. Τόσο οι ώρες όσο και τα λεπτά πρέπει να καταλαμβάνουν από 2 θέσεις και αν είναι μονοψήφια να ξεκινούν από 0. Δείτε τα ενδεικτικά αρχεία εξόδου για περισσότερα παραδείγματα.

Ολοκληρώστε αυτό το στάδιο και βεβαιωθείτε ότι λειτουργεί σωστά πριν προχωρήσετε στο επόμενο.

Στάδιο 5: Υλοποίηση διαγραφής ραντεβού (**)

Σε αυτό το στάδιο θα προσθέσετε κώδικα στα stubs που ορίσατε για τις διαγραφές ραντεβού.

Για τη διαγραφή με βάση την κατηγορία, το πρόγραμμα:

Ζητά από το χρήστη να προσδιορίσει την κατηγορία του ραντεβού εκτυπώνοντας το μενού:

```
Select category:
0. MEETING
1. CLASS
2. SKYPE CALL
-->
```

Πριν από τις ακέραιες τιμές υπάρχει χαρακτήρας tab.

Διαβάζει την κατηγορία, βρίσκει όλα τα ραντεβού αυτής της κατηγορίας στο ημερολόγιο και τα διαγράφει.

Για τη διαγραφή με βάση την περιγραφή, το πρόγραμμα:

Ζητά από το χρήστη να προσδιορίσει την περιγραφή του ραντεβού εκτυπώνοντας το μήνυμα:

Enter short description:

ακολουθούμενο από ένα **χαρακτήρα κενό** (space). Διαβάζει την περιγραφή βρίσκει όλα τα ραντεβού αυτής της κατηγορίας στο ημερολόγιο και τα διαγράφει.

Σε κάθε περίπτωση, αν έχει ενεργοποιηθεί το debug, τότε για κάθε ραντεβού που σβήνει εκτυπώνει το μήνυμα:

Removed appointment on day D.

ακολουθούμενο από **χαρακτήρα αλλαγής γραμμής**. D είναι η ημέρα (μετρώντας από το 1).

Επίσης, σε κάθε περίπτωση, βεβαιωθείτε ότι ο κώδικας σας θα μπορεί αργότερα να προσθέσει ένα νέο ραντεβού σε θέση που διαγράφηκε προηγούμενα.

Στάδιο 6: Υλοποίηση απελευθέρωσης δυναμικά δεσμευμένης μνήμης (*)

Σε αυτό το στάδιο προσθέστε κώδικα στο stub που ορίσατε για την απελευθέρωση όλης της δυναμικά δεσμευμένης μνήμης για τα ραντεβού του ημερολογίου. Θα καλείτε αυτή τη συνάρτηση σε κάθε περίπτωση που πρόκειται να τερματίσει το πρόγραμμα: είτε επειδή το ζήτησε ο χρήστης, είτε επειδή ανιχνεύθηκε κάποιο σοβαρό λάθος, όπως αποτυχία της malloc. Ειδικά σε περίπτωση αποτυχίας της malloc, το πρόγραμμα πρέπει να τερματίζει έχοντας εκτυπώσει ένα σχετικό μήνυμα.

Εδώ ολοκληρώνεται το μέρος Α. Κάντε τελικούς ελέγχους ορθότητας χρησιμοποιώντας τα αρχεία ελέγχου που θα σας δώσουμε.

Όταν είστε σίγουροι ότι αυτό το μέρος λειτουργεί σωστά, αποθηκεύστε ένα αντίγραφο του προγράμματός σας με άλλο όνομα και συνεχίστε στο δεύτερο μέρος.

Μέρος B: Πολλά ραντεβού τη μέρα.

Εισαγωγή

Στο δεύτερο μέρος της εργασίας θα προσθέσετε κώδικα στο πρόγραμμά σας ώστε να επεκτείνετε τη λειτουργία του ημερολογίου σας. Αυτή τη φορά θέλουμε να μπορούμε να αποθηκεύσουμε πολλαπλά ραντεβού σε μια μέρα. Για να το κάνουμε αυτό, θα πρέπει κάθε ραντεβού να είναι κόμβος σε μια λίστα από ραντεβού. Το ημερολόγιό μας εξακολουθεί να είναι πίνακας από δείκτες, αλλά τώρα κάθε δείκτης δείχνει στον πρώτο κόμβο μιας λίστας από ραντεβού. Η λίστα θα είναι διπλά διασυνδεδεμένη, χωρίς τερματικό.

Επειδή γράψαμε το πρώτο μέρος προσεκτικά, με καλή χρήση δομών και συναρτήσεων, θα είναι σχετικά εύκολο να κάνουμε τις απαιτούμενες αλλαγές.

Σε πολλά σημεία θα πρέπει να αποφασίσετε εσείς αν και τι είδους αλλαγές χρειάζεται να γίνουν στον ήδη υπάρχοντα κώδικα.

Στάδιο 1: Δομές και αρχικοποίηση ημερολογίου (*)

Εφόσον κάθε ραντεβού θα είναι ένας κόμβος της λίστας, προσθέστε στη δομή για το ραντεβού δύο πεδία:

- Ένα δείκτη προς το επόμενο ραντεβού της λίστας
- Ένα δείκτη προς το προηγούμενο ραντεβού της λίστας.

Εξετάστε αν χρειάζεται να κάνετε αλλαγές στην αρχικοποίηση του ημερολογίου, κι αν ναι, κάντε τις.

Στάδιο 2: Δημιουργία και εισαγωγή νέου ραντεβού (**)

Σε αυτό το στάδιο, θα πρέπει να εισάγετε το νέο ραντεβού στη λίστα από ραντεβού για τη συγκεκριμένα ημέρα (ή ημέρες αν έχουμε μη-μηδενικό διάστημα επανάληψης). Το ραντεβού πρέπει να εισάγεται στη λίστα με τέτοιο τρόπο ώστε η λίστα να είναι κάθε στιγμή ταξινομημένη με βάση την ώρα έναρξης. Δε χρειάζεται να ελέγχετε αν υπάρχει επικάλυψη μεταξύ διαφορετικών ραντεβού.

Στάδιο 3: Υπόλοιπες συναρτήσεις (**)

Προσθέστε κώδικα όπου χρειάζεται στις υπόλοιπες συναρτήσεις ώστε η κάθε λειτουργία να είναι κατάλληλη για όλα τα ραντεβού. Για παράδειγμα, στην αφαίρεση ανά κατηγορία θα πρέπει να αφαιρεθούν όλα τα ραντεβού (σε όλες οι λίστες όλων των ημερών) που ανήκουν στη δεδομένη κατηγορία.

Στάδιο 4: Τελικός έλεγχος ορθότητας

Το πρόγραμμά σας πρέπει να λειτουργεί σωστά και να εκτυπώνει όλα τα μηνύματα και αποτελέσματα με τον τρόπο που σας περιγράφουμε. Θα σας δώσουμε ενδεικτικά αρχεία εισόδου και εξόδου.

Πρέπει το πρόγραμμα που θα μας παραδώσετε να παράγει έξοδο που δεν έχει διαφορές από τη δική μας.

Θα παραδώσετε ΜΟΝΟ τη λύση σας για το δεύτερο μέρος.

Αρχείο προς παράδοση: hw4.c

Πώς να παραδώσετε τη δουλειά σας

Προσθέστε σε σχόλια στην αρχή του κάθε αρχείου με κώδικα τα πλήρη ονόματα και ΑΕΜ των μελών της ομάδας. Παρακαλούμε να γράφετε τα σχόλια ΜΟΝΟ με λατινικούς χαρακτήρες.

Κατασκευάστε ένα φάκελο με όνομα `hw4_epwnumero1_AEM1_epwnumero2_AEM2`

Αντιγράψτε μέσα σε αυτόν το `hw4.c`

Συμπιέστε το φάκελο `hw4_epwnumero1_AEM1_epwnumero2_AEM2` ως `.tar.gz`

Στείλτε email:

- στη διεύθυνση **CE120LAB@gmail.com**
- αντίγραφο (CC) στον άλλο μέλος της ομάδας σας
- θέμα (subject) **hw4**
- και επικολλημένο αρχείο το `hw4_epwnumero1_AEM1_epwnumero2_AEM2.tgz`

Ασκήσεις που έχουν λάθος format ή δεν έχουν σταλεί σωστά έχουν penalty στο βαθμό