

## lab12: Δυναμική διαχείριση μνήμης

### Προετοιμασία:

1. Πλοηγηθείτε στον κατάλογο lab12 ο οποίος πρέπει να βρίσκεται μέσα στον κατάλογο ce120.
2. Ολοκληρώστε τις ασκήσεις της εβδομάδας. Τα αρχεία που θα γράψετε πρέπει να αποθηκευτούν στον κατάλογο lab12.

### Θέματα εργασίας:

- Λίστες
- Δυναμική διαχείριση μνήμης
- struct
- Δείκτες
- Συναρτήσεις
- Συμβολοσειρές

### Αποστολή Προγράμματος για σχολιασμό και βαθμολόγηση

- Πλοηγηθείτε στον κατάλογο lab12. Θα πρέπει να βρίσκεται μέσα το αρχείο lab12.c
- Κατασκευάστε να κατάλογο με όνομα lab12\_επώνυμο1\_AEM1\_επώνυμο2\_AEM2 (ΜΕ ΛΑΤΙΝΙΚΟΥΣ ΧΑΡΑΚΤΗΡΕΣ) όπου επώνυμο1, AEM1 αναφέρονται στο ένα μέλος της ομάδας και επώνυμο2, AEM2 στο δεύτερο μέλος. Για παράδειγμα, αν τα δύο μέλη της ομάδας ήταν οι Ντουφεξή με AEM 01234 και Θάνος με AEM 05678, θα γράφατε:

```
mkdir lab12_Doufexi_01234_Thanos_05678
```

- Αντιγράψτε το αρχείο C μέσα στον κατάλογο. lab12\_επώνυμο1\_AEM1\_επώνυμο2\_AEM2
- Πακετάρετε τον κατάλογο (compress here as tar.gz).
- Το ένα μέλος της ομάδας:
  - μπαίνει στο λογαριασμό του στο eclass,
  - πηγαίνει στην ενότητα Εργασίες,
  - κάνει click στην εργασία lab12 για το συγκεκριμένο τμήμα
  - **ΠΡΟΣΟΧΗ!!!!** Στο πεδίο Ομάδα Χρηστών επιλέγει τη σωστή ομάδα (με βάση τα ονόματα των μελών)
  - Ανεβάζει το .tar.gz αρχείο μέσω του Browse **και πατά Αποθήκευση.**
  - **ΠΡΟΣΟΧΗ!!!!** Επιβεβαιώνει ότι ανέβασε το σωστό αρχείο.

## Άσκηση

Το πρόγραμμα που θα γράψετε για την άσκηση πρέπει να το αποθηκεύσετε στο αρχείο με όνομα `lab12.c` (σας δίνεται έτοιμος σκελετός) μέσα στον κατάλογο `lab12`. Για να μεταγλωττίσετε το πρόγραμμα σας αρκεί να γράψετε

```
gcc -Wall -g lab12.c -o lab12 -L. -lcolor
```

Σας δίνεται έτοιμο το αρχείο `lab12.h` το οποίο περιέχει το `struct color` το οποίο αντιπροσωπεύει τον κόμβο μιας διπλά διασυνδεδεμένης μη-κυκλικής λίστας με τερματικό κόμβο στην αρχή της. Κάθε κόμβος της λίστας περιγράφει ένα `RGB` χρώμα και έχει τα εξής πεδία:

- `name (char *)`: Ένας δείκτης σε μία δυναμικά δεσμευμένη περιοχή μνήμης στην οποία αποθηκεύεται το όνομα του χρώματος.
- `red, green, blue (int)`: Αντιπροσωπεύει τις τιμές του κόκκινου, πράσινου και μπλε στο σύστημα `RGB`.

Στη βιβλιοθήκη `libcolors.a` υλοποιούνται οι εξής συναρτήσεις:

- `init_colors`: Δημιουργεί μία λίστα από χρώματα και επιστρέφει την κεφαλή της λίστας.
- `print_colors`: Εκτυπώνει τα περιεχόμενα της λίστας.
- `find_color`: Εντοπίζει τον κόμβο που έχει όνομα `name` και επιστρέφει τη διεύθυνση του εφόσον τον εντοπίσει ή `NULL` εάν δεν υπάρχει κόμβος με το συγκεκριμένο πεδίο `name`.

Σας δίνεται ο σκελετός ενός προγράμματος το οποίο στην τελική του μορφή πρέπει να υλοποιεί τα εξής βήματα:

1. Με τη βοήθεια της συνάρτησης `init_colors` δημιουργεί μία διπλά διασυνδεδεμένη μη-κυκλική λίστα με τερματικό κόμβο στην αρχή της που περιέχει κόμβους τύπου `struct color`.

Καλέστε τη συνάρτηση `init_colors` στο σημείο του προγράμματος που υποδεικνύουν τα σχόλια.

2. Με τη βοήθεια της συνάρτησης `print_colors` εκτυπώνει τα περιεχόμενα της λίστας.

Καλέστε τη συνάρτηση `print_colors` στο σημείο του προγράμματος που υποδεικνύουν τα σχόλια.

3. Εκτυπώνει το μήνυμα "**Enter color:** ", διαβάζει το όνομα ενός χρώματος της λίστας από το πληκτρολόγιο και στη συνέχεια ψάχνει να βρει τον κόμβο της λίστας που περιέχει το παραπάνω όνομα. Εάν δεν το βρει εκτυπώνει "**Invalid color!**" ακολουθούμενο από χαρακτήρα αλλαγής γραμμής και επαναλαμβάνει το βήμα.

Καλέστε τη συνάρτηση `find_color` στο σημείο του προγράμματος που υποδεικνύουν τα σχόλια.

4. Δημιουργεί ένα νέο κόμβο με όνομα "`shadow-name`", όπου `name` το όνομα που διάβασε στο προηγούμενο βήμα. Οι τιμές `RGB` για τον κόμβο αυτό είναι 0.85 των τιμών του κόμβου που εντόπισε στο βήμα 3. Ενθέτει το νέο κόμβο στην αρχή της

λίστας.

5. Δημιουργεί ένα νέο κόμβο με όνομα "**dark-name**", όπου **name** το όνομα που διάβασε στο προηγούμενο βήμα. Οι τιμές **RGB** για τον κόμβο αυτό είναι 0.5 των τιμών του κόμβου που εντόπισε στο βήμα 3. Ενθέτει το νέο κόμβο στην αρχή της λίστας.
6. Διαγράφει τον κόμβο που εντόπισε στο βήμα 3 ελευθερώνοντας όλη τη δυναμικά δεσμευμένη μνήμη.
7. **(προαιρετικό βήμα)** Εκτυπώνει το μήνυμα "**Enter red color range: (from|to)**" και διαβάζει δύο ακέραιους αριθμούς διαχωριζόμενους από τον χαρακτήρα |. Οι ακέραιοι που διάβασε αντιπροσωπεύουν ένα άνω και κάτω όριο φωτεινότητας. Στη συνέχεια διαγράφει από τη λίστα όλους τους κόμβους που περιέχουν τιμές φωτεινότητας για το κόκκινο χρώμα εκτός του παραπάνω εύρους τιμών.  
Για το παραπάνω βήμα, θεωρείστε ότι ο αριθμός που δίνεται πρώτος είναι πάντοτε μικρότερος του αριθμού που δίνεται δεύτερος.
8. **(προαιρετικό βήμα)** Εκτυπώνει ξανά τα περιεχόμενα της λίστας.

**Σημείωση:** Συνιστάται να φτιάξετε συναρτήσεις για την υλοποίηση επιμέρους λειτουργιών του προγράμματος σας. Ειδικότερα είναι υποχρεωτική η υλοποίηση συναρτήσεων για την ένθεση ενός νέου κόμβου στη λίστα και τη διαγραφή ενός υφιστάμενου κόμβου από τη λίστα.