

Γενικά σχόλια lab12

ΜΕΤΑΓΛΩΤΤΙΣΗ/ΣΤΟΙΧΙΣΗ/ΚΕΝΑ:

Ισχύουν τα ίδια με προηγούμενες εργασίες.

ΚΛΗΣΕΙΣ:

Καλούμε την `init_colors` και αναθέτουμε το αποτέλεσμα στη μεταβλητή `head`.

Περνάμε το `head` ως παράμετρο στην `print_colors`.

Καλούμε τη `find_color` με παράμετρο το χρώμα που διαβάστηκε και αποθηκεύουμε το αποτέλεσμα στη μεταβλητή `color`.

ΝΕΟΙ ΚΟΜΒΟΙ:

1. `malloc` για τη δημιουργία του κόμβου.
2. `malloc` για τη δημιουργία του απαραίτητου χώρου για την αποθήκευση της συμβολοσειράς στο πεδίο `name`. Για το μέγεθος στη `malloc` χρησιμοποιούμε το μέγεθος του χρώματος συν το μέγεθος του "shadow-" ή "dark-" συν 1 για το '\0'. Στη συνέχεια, κατάλληλη χρήση `strcpy/strcat` για τα ονόματα των νέων κόμβων.
3. Εισαγωγή του κόμβου στην 1η θέση της λίστας θέτοντας κατάλληλα τους δείκτες `next`, `prev` του κόμβου που εισάγεται, τον δείκτη `next` του προηγούμενου κόμβου και τον δείκτη `prev` του επόμενου κόμβου.

Παραλλαγή: Δοκιμάστε να κάνετε την άσκηση με εισαγωγή σε λίστα που είναι ταξινομημένη με βάση το πεδίο `name` (λεξικογραφικά κατά αύξουσα σειρά). Το ενδιαφέρον σε αυτή την περίπτωση είναι ότι πρέπει να κάνετε ειδικό έλεγχο για την περίπτωση που εισάγετε κόμβο στο τέλος της λίστας (διότι εάν το `next` ενός κόμβου είναι `NULL`, τότε δεν επιτρέπεται να προσπελάσετε το `next->prev` γιατί θα λάβετε `segmentation fault`).

ΔΙΑΓΡΑΦΗ ΚΟΜΒΟΥ:

Περιμένουμε να γράψετε μια συνάρτηση που παίρνει ως παράμετρο ένα κόμβο και τον διαγράφει από την λίστα. Η διαγραφή αποτελείται από δύο πράξεις. Την αφαίρεση του κόμβου από τη λίστα και **ΜΕΤΑ** την αποδέσμευση της μνήμης γι αυτόν. Για παράδειγμα, αν έχετε μια λίστα με στοιχεία `A <-> B <-> C` και θέλετε να σβήσετε τον κόμβο που περιέχει το `B`, τότε πρώτα θα θέσετε το `A->next` στο `C` και το `C->prev` στο `A` ώστε να αφαιρεθεί το `B` από τη λίστα, και μετά θα κάνετε `free` το `B`. Επιπλέον, είναι λάθος να κάνετε `free(B->next)`; `free(B->prev)`; γιατί αυτό θα έχει ως αποτέλεσμα να γίνουν `free` τα `C` και `A` αντίστοιχα.

Άρα, στη συγκεκριμένη άσκηση:

Διαγραφή: Η διαγραφή γίνεται θέτοντας ανάλογα τα `prev/next` του προηγούμενου και του επόμενου κόμβου. **Προσοχή:** θα πρέπει να λάβετε ιδιαίτερη μέριμνα για την περίπτωση που ο κόμβος `color` που θα διαγράψετε βρίσκεται στο τέλος της λίστας (σε αυτή την περίπτωση ο επόμενός του είναι `NULL`).

Απελευθέρωση μνήμης: Στη συνέχεια, στην ίδια συνάρτηση, κάνετε free ΠΡΩΤΑ ότι πεδία έχουν δεσμευτεί δυναμικά (δηλαδή το πεδίο name) και ΜΕΤΑ τον ίδιο τον κόμβο.

ΔΙΑΓΡΑΦΗ ΚΟΜΒΩΝ (προαιρ.):

Καλείτε τη συνάρτηση που γράψατε προηγουμένα μέσα σε μία επανάληψη από τη κεφαλή έως το τέλος προκειμένου να διαγράψετε τους κόμβους που πληρούν τα συγκεκριμένα κριτήρια φωτεινότητας του κόκκινου χρώματος.

Προσοχή σε λάθος της μορφής:

```
for (current = head->next; current != NULL; current = current->next) {
    free(current->name);
    free(current);
}
```

Αυτό είναι λάθος γιατί σε μια επανάληψη κάνει free τον κόμβο current και πριν την επόμενη επανάληψη κάνει την πράξη current = current->next. Όμως το current->next δεν είναι πια έγκυρο γιατί το current έχει γίνει free.

Ο σωστός τρόπος είναι με χρήση βοηθητικής μεταβλητής στην οποία αναθέτουμε το current->next πριν γίνει free το current:

```
for (current = head->next; current != NULL; current = nextnode) {
    nextnode = current->next;
    free(current->name);
    free(current);
}
```