

## Παρατηρήσεις κι επεξηγήσεις για το lab9

Για μεταγλώττιση, στοίχιση/κενά, ονόματα και έξοδο ισχύουν τα ίδια με προηγούμενες ασκήσεις.

### Συναρτήσεις:

- Η συνάρτηση εκτύπωσης πρέπει να παίρνει ως παράμετρο τον πίνακα χρηστών και να έχει void return type.
- Η συνάρτηση εύρεσης κοινών φίλων πρέπει να παίρνει ως παραμέτρους δύο strings (είτε ως char[] είτε ως char\*), τον πίνακα χρηστών κι ένα αρχικά άδειο πίνακα ακεραίων μεγέθους MAX\_USERS. Επιστρέφει ακέραιο. Η τιμή που επιστρέφει είναι ουσιαστικά το πλήθος ακεραίων που έχουν αποθηκευτεί στην τελευταία παράμετρο.
  - Η τιμή επιστροφής πρέπει να ελέγχεται στη main κι αν είναι -1 να μην εκτυπώνεται τίποτα, διαφορετικά να διατρέχεται ο πίνακας (με βάση αυτό το πλήθος) στον οποίο βρίσκονται οι κοινοί φίλοι και να εκτυπώνει τα ονόματά τους.
- Η συνάρτηση εύρεσης του πιο δημοφιλούς χρήστη πρέπει να παίρνει ως παραμέτρους τον πίνακα χρηστών κι ένα δείκτη σε ακέραιο.
  - Στη main ορίζουμε μια ακέραια μεταβλητή και περνάμε τη διεύθυνσή της ως παράμετρο στη συνάρτηση. Επιστρέφοντας από τη συνάρτηση, η μεταβλητή θα έχει πάρει την κατάλληλη τιμή.
  - Η τιμή που επιστρέφει η συνάρτηση αποθηκεύεται σε δεύτερη ακέραια μεταβλητή.

### PRINT:

Το ενδιαφέρον εδώ είναι η εκτύπωση των φίλων/followers κάθε χρήστη. Για κάθε χρήστη i θέλουμε να διατρέξουμε τον πίνακα φίλων/followers, και για κάθε θέση αυτού του πίνακα που αντιστοιχεί σε έγκυρο χρήστη να ελέγξουμε αν περιέχει YES. Σε αυτή την περίπτωση, πηγαίνουμε στην αντίστοιχη θέση στον πίνακα users και εκτυπώνουμε το όνομα που βρίσκεται εκεί. Δηλαδή:

```
for (i=0; i<MAX_USERS; i++) {
    if (users[i].name[0]!='\0') {
        printf("%s:", users[i].name);
        for (j=0; j<MAX_USERS; j++) {
            if (users[i].friends[j] == YES) {
                printf(" %s", users[j].name);
            }
        }
        printf("\n");
    }
}
```

Παρατηρήστε επίσης πως ο έλεγχος για το αν ένας χρήστης είναι έγκυρος πρέπει να είναι έλεγχος για το αν το όνομά του ξεκινά από '\0'.

Δεν είναι σωστό να γίνεται έλεγχος για το αν περιέχει NO σε κάποια θέση του πεδίου friends/followers γιατί αυτό μπορεί να συμβαίνει και για έγκυρους χρήστες. Θα μπορούσε να γίνεται έλεγχος για το αν ΟΛΕΣ οι θέσεις του friends/followers έχουν NO αλλά αυτό είναι πιο πολύπλοκο από ένα απλό έλεγχο για '\0' στο όνομα.

Ακόμη πιο σοβαρό λάθος είναι ο μοναδικός έλεγχος για το εξωτερικό loop να είναι users[i].name[0]!='\0' χωρίς να υπάρχει έλεγχος για το i<MAX\_USERS. Σε αυτή την περίπτωση, αν ο πίνακας χρηστών είναι γεμάτος χρήστες, το loop θα συνεχίσει να εκτελείται ακόμα κι όταν το i υπερβεί το μέγεθος του πίνακα.

Επιπλέον, δεν πρέπει να υποθέτετε ότι όλοι οι έγκυροι χρήστες είναι σε διαδοχικές θέσεις. Μπορεί να υπάρχουν και άκυροι χρήστες ανάμεσα σε έγκυρους.

### COMMON FRIENDS [Πέμπτη]:

Ο πίνακας ακεραίων στον οποίο θα αποθηκευτούν οι κοινοί φίλοι πρέπει να αρχικοποιηθεί σε κάτι "άκυρο", για παράδειγμα -1. Το 0 δεν είναι καλή επιλογή γιατί υπονοεί ότι δύο χρήστες έχουν κοινό φίλο τον χρήστη users[0].

Διατρέχουμε τον πίνακα χρηστών μία φορά για να δούμε αν υπάρχουν και τα δύο ονόματα. Επειδή θα χρειαστούμε τις θέσεις τους, θα πρέπει να έχουμε αρχικά δηλώσει δύο ακέραιες μεταβλητές για αυτές και να τις έχουμε αρχικοποιήσει σε μια άκυρη τιμή, για παράδειγμα -1. Αν βρούμε κάποιο από τα ονόματα, κρατάμε τη θέση του στην αντίστοιχη μεταβλητή.

Αφού έχει ολοκληρωθεί το loop (και μόνο τότε!) ελέγχουμε αν και οι δύο μεταβλητές περιέχουν τιμές διάφορες του -1. Σε αυτή την περίπτωση προχωράμε, αλλιώς επιστρέφουμε άμεσα -1.

Εφόσον προχωρήσουμε, διατρέχουμε "παράλληλα" τις λίστες φίλων των δύο χρηστών και ελέγχουμε αν και οι δύο έχουν YES σε αντίστοιχες θέσεις. Σε αυτή την περίπτωση αποθηκεύουμε τις θέσεις αυτές σε διαδοχικά κελιά του πίνακα ακεραίων που περάσαμε ως παράμετρο. Η λογική, όσον αφορά το γέμισμα του πίνακα, είναι παρόμοια με αυτή του lab9.

Στο τέλος η συνάρτηση επιστρέφει το πλήθος των κελιών του πίνακα ακεραίων στα οποία έχουμε γράψει.

### MOST POPULAR [Παρασκευή]:

Τυπικός αλγόριθμος εύρεσης maximum. Αρχικοποιούμε μια μεταβλητή max σε -1 ή 0. Σε αυτή θα κρατάμε κάθε φορά το τρέχον μέγιστο πλήθος από followers. Παράλληλα, χρειαζόμαστε μια ακέραια μεταβλητή στην οποία θα αποθηκεύουμε τη θέση του χρήστη που αντιστοιχεί στο τρέχον μέγιστο πλήθος από followers.

Διατρέχουμε τον πίνακα χρηστών και για κάθε χρήστη που είναι έγκυρος διατρέχουμε τον πίνακα από followers και μετράμε τα YES. Αν το πλήθος είναι καλύτερο από το τρέχον μέγιστο, ανανεώνουμε το τρέχον μέγιστο, καθώς και την αντίστοιχη θέση. Στο τέλος, αποθηκεύουμε το μέγιστο στον προορισμό του δείκτη που περάσαμε ως παράμετρο και επιστρέφουμε τη θέση.