

## Παρατηρήσεις πάνω στο lab3

### Μεταγλώττιση

Δεν πρέπει να αγνοείτε τις προειδοποιήσεις του compiler (warnings) γιατί επισημαίνουν πιθανά λάθη στον κώδικά σας, τα οποία ναί μεν δεν επηρεάζουν την παραγωγή εκτελέσιμου, αλλά σχεδόν πάντα επηρεάζουν την ορθότητα και τα αποτελέσματα του προγράμματός σας. Δείτε για παράδειγμα το παρακάτω warning:

```
lab3.c:26:6: warning: suggest parentheses around assignment used as truth value [-Wparentheses]
   26 |   if (game_month = this_month) {
      |           ^~~~~~
```

Το λάθος που έχει γίνει εδώ είναι ότι χρησιμοποιείται τελεστής ανάθεσης τιμής (=) αντί για τελεστή σύγκρισης ισότητας (==). Αν δε διορθωθεί το λάθος, τότε κατά την εκτέλεση θα αποτιμηθεί η έκφραση ανάθεσης τιμής: επειδή το this\_month είναι πάντα θετικός αριθμός, η έκφραση είναι πάντα αληθής και κατά συνέπεια το σώμα της if εκτελείται πάντα.

Ενα άλλο συχνό λάθος ήταν το παρακάτω:

```
lab3.c:26:59: warning: suggest parentheses around '&&' within '||' [-Wparentheses]
   26 |   if (game_month < this_month || game_month == this_month && game_day <
      |                                           ^~~~~~
      |                                           this_day)) {
```

Εδώ το πρόβλημα είναι ότι δεν έχει δοθεί προσοχή στην προτεραιότητα των τελεστών || και &&. Το && έχει μεγαλύτερη προτεραιότητα από το || και ο compiler μας ενημερώνει ότι καλό θα ήταν να χρησιμοποιήσουμε παρενθέσεις γύρω από το ζεύγος συγκρίσεων (ροζ) για να είναι πιο σαφής η σειρά των συγκρίσεων.

### Τελεστές

Δώστε ιδιαίτερη προσοχή στους τελεστές και πότε χρησιμοποιείται & έναντι &&.

Το && είναι λογικός τελεστής. Αντιμετωπίζει τους τελεστέους του ως λογικές εκφράσεις και το αποτέλεσμα είναι η τιμή 0 (ψευδές) ή 1 (αληθές). Για παράδειγμα, αν int x=5, y=2; τότε στην έκφραση x && y το x ερμηνεύεται ως "αληθές" (διότι δεν είναι ίσο με το μηδέν), το y ομοίως, επομένως το αποτέλεσμα της έκφρασης είναι αληθές (1). Για τις συγκεκριμένες τιμές των x, y, η έκφραση x&&y μέσα σε μια συνθήκη if έχει ως αποτέλεσμα να εκτελεστεί το σώμα της if.

Το & είναι bitwise τελεστής. Στην έκφραση x & y το & δρα σε επίπεδο bit των x και y εκτελώντας την πράξη "δυναμικό και" ανάμεσα τους. Εφόσον το x είναι ίσο με 5 (στο δυαδικό 101) και το y είναι ίσο με 2 (στο δυαδικό 010), τότε το x & y παίρνει την τιμή 0 (000 διότι 1&0: 0, 0&1: 0, 1&0: 0). Επομένως σε αυτή την περίπτωση η έκφραση x&y μέσα σε μια συνθήκη if έχει ως αποτέλεσμα να ΜΗΝ εκτελεστεί το σώμα της if.

### Character literals

Αν στο πρόγραμμα σας χρειάζεται να χρησιμοποιήσετε κάποιο character literal (όπως το 'A' στο lab3 ή το 'm' στο lab2, τότε να χρησιμοποιείτε τον ίδιο τον χαρακτήρα ('A') κι όχι την ASCII τιμή του (65 ή 0x41), για να είναι πιο κατανοητός ο κώδικας.

### Στοίχιση

Κάθε φορά που ανοίγετε άγκιστρο, ο κώδικας που ακολουθεί πρέπει να είναι ένα tab πιο δεξιά, μέχρι να κλείσει το άγκιστρο. Διαβάστε το φυλλάδιο μορφοποίησης κώδικα!

## switch

Σε μια εντολή switch, εάν θέλετε να κάνετε το ίδιο πράγμα για δύο ή περισσότερες επιλογές, τότε μπορείτε να εκμεταλλευτείτε το λεγόμενο fall-through effect: Όσο δεν συναντάται break η εκτέλεση συνεχίζεται με το επόμενο σώμα. Για παράδειγμα:

```
switch (row) {
    case 'F':
    case 'E':
        printf("10%% off.\n");
        break;
}
```

Επι τη ευκαιρία, παρατηρήστε τη στοίχιση της switch.

## Υπολογισμός προγενέστερης ημερομηνίας και ώρας

Για τον υπολογισμό της προγενέστερης ημερομηνίας και ώρας, πρέπει να συγκριθούν πρώτα οι μήνες. Αν δεν είναι ίσοι, τότε είναι ξεκάθαρο ποιο γεγονός γίνεται πρώτο και δε χρειάζεται να γίνει άλλος έλεγχος. Αν είναι ίσοι, τότε πρέπει να συγκριθούν οι ημέρες. Αν δεν είναι ίσες, τότε είναι ξεκάθαρο ποιο γεγονός γίνεται πρώτο και δε χρειάζεται να γίνει άλλος έλεγχος. Αν είναι ίσες, τότε πρέπει να συγκριθούν οι ώρες.

Μια εναλλακτική ιδέα είναι να μετατραπούν σε κάθε περίπτωση ο μήνας, η ημέρα και η ώρα σε συνολικές ώρες από την αρχή του έτους. Εδώ μπορεί να χρησιμοποιηθεί η υπόθεση ότι κάθε μήνας έχει 30 ημέρες και φυσικά το ότι κάθε ημέρα έχει 24 ώρες. Με αυτόν τον τρόπο, η εύρεση του προγενέστερου γεγονότος είναι μια απλή if.

## Υποβολή

Να διαβάζετε κάθε φορά προσεκτικά τις οδηγίες υποβολής της άσκησης. Στο lab2 χρειάστηκε να υποβάλετε αρχείο zip γιατί οι ασκήσεις ήταν δύο. Στο lab3 έπρεπε να υποβάλετε το lab3.c. Εάν δεν υποβάλετε σωστό είδος αρχείου, το autolab δε μπορεί να βαθμολογήσει την άσκηση σας.

Μέχρι και το lab3 θα δικαιολογήσουμε λάθη στην υποβολή, αλλά από το lab4 και μετά αυτό δε θα το κάνουμε.