

## lab4: Bitwise τελεστές, συνθήκες

### Πριν ξεκινήσετε...

Πατήστε πάνω στο σύνδεσμο "Download handout" και ανοίξτε (open) το αρχείο `lab4.tar.gz`. Θα δείτε το παράθυρο της εφαρμογής Ark στο οποίο εμφανίζονται τα περιεχόμενα του πακέτου `lab4.tar.gz`. Σύρετε (drag & drop) τον κατάλογο `lab4` από το παράθυρο του Ark μέσα στο φάκελο `ce120` που βρίσκεται στο home directory σας. Θα δουλέψετε μέσα στον κατάλογο `lab4` που μόλις μεταφέρατε.

### Έλεγχος ασκήσεων

Όταν ολοκληρώσετε τις ασκήσεις σας, γράψτε στο τερματικό σας την εντολή `make`. Θα ελέγξει τα αποτελέσματά σας ώστε να επιβεβαιώσετε ότι είναι σωστά πριν υποβάλετε τα προγράμματα στο Autolab.

### Υποβολή ασκήσεων

1. **Πριν παραδώσετε τις ασκήσεις πρέπει να έχετε δημιουργήσει την ομάδα σας στο Autolab!**
2. **Αντιγράψτε** το `lab4a.c` στον κατάλογο `lab4submit` που βρίσκεται μέσα στο `lab4`.
  - Δεν πρέπει να υπάρχουν άλλα αρχεία σε αυτόν τον κατάλογο.
3. Κάντε **δεξί κλικ** στον κατάλογο `lab4submit` και επιλέξτε `Compress` → `Here as tar.gz`
4. Θα έχει δημιουργηθεί ένα αρχείο με όνομα `lab4submit.tar.gz` το οποίο ένα μέλος της ομάδας (δεν έχει σημασία ποιο) **υποβάλει** στο Autolab.

Αν το Autolab σας δώσει μηδενικό σκορ σε κάποιο από τα τεστ, μπορείτε να δείτε ποιο ακριβώς ήταν αυτό στον κατάλογο `tests` που βρίσκεται μέσα στον κατάλογο `lab4` που αποσυμπίεσατε στην αρχή του εργαστηρίου. Σε κάθε τεστ αντιστοιχούν δύο αρχεία με ονόματα της μορφής `X_in_Y` και `X_out_Y`, όπου το `X` έχει την τιμή 'a' για την άσκηση 1 ή 'b' για την άσκηση 2 ενώ το `Y` είναι ο αύξων αριθμός του τεστ. Κάθε αρχείο `X_in_Y` περιέχει την είσοδο που δίνεται στο πρόγραμμα και κάθε αρχείο `X_out_Y` την αντίστοιχη έξοδο για αυτή την είσοδο.

Σημειώστε ότι, πέρα από τα τεστ που σας δίνουμε έτοιμα, είναι δική σας ευθύνη να σχεδιάσετε και να εκτελέσετε όποιο άλλο τεστ κρίνετε απαραίτητο ώστε να ελέγξετε την ορθή λειτουργία των προγραμμάτων σας.

Προσοχή: Τα προγράμματά σας πρέπει να είναι πάντα σωστά στοιχισμένα, να περιέχουν κατάλληλα σχόλια, να έχουν περιγραφικά ονόματα μεταβλητών και σωστούς τύπους, και η έξοδός τους να συμμορφώνεται πλήρως με τις προδιαγραφές.

## Άσκηση 1

Το πρόγραμμα που θα γράψετε για την άσκηση 1 πρέπει να το αποθηκεύσετε σε αρχείο με όνομα **lab4a.c** μέσα στον κατάλογο lab4 που προέκυψε από την αποσυμπίεση του lab4.tar.gz που κατεβάσατε από το Autolab.

Σε αυτή την άσκηση θα δείτε ένα παράδειγμα χρήσης bitwise τελεστών για την κωδικοποίηση διαφορετικών ειδών δεδομένων σε περιορισμένο αριθμό από bits.

Το πρόγραμμά σας θα πάρει ως είσοδο ένα δεκαεξαδικό αριθμό μεγέθους 32 bits (τύπου unsigned int) στα οποία αποθηκεύονται πληροφορίες εισιτηρίου για ένα ποδοσφαιρικό αγώνα.

Η κωδικοποίηση έχει γίνει ως εξής, ξεκινώντας από το πιο σημαντικό bit:

- Στα 6 πιο σημαντικά bits έχει αποθηκευτεί ένας ακέραιος που αντιστοιχεί στην πύλη και μπορεί να πάρει τιμές από 1 έως και 35.
- Στα επόμενα 5 bits έχει αποθηκευτεί η σειρά ως "απόσταση" από το γράμμα 'A'. Για παράδειγμα, αν η θέση βρίσκεται στη σειρά 'G' τότε έχει αποθηκευτεί ο αριθμός 6 γιατί το 'G' απέχει 6 γράμματα από το 'A'.
- Στα επόμενα 7 bits έχει αποθηκευτεί ο αριθμός της θέσης, ο οποίος μπορεί να πάρει τιμές από 1 έως και 120.
- Στα επόμενα 4 bits έχει αποθηκευτεί ο μήνας, ο οποίος μπορεί να πάρει τιμές από 1 έως και 12.
- Στα επόμενα 5 bits έχει αποθηκευτεί η ημέρα, η οποία μπορεί να πάρει τιμές από 1 έως και 31.
- Στα επόμενα 5 bits έχει αποθηκευτεί η ώρα που ξεκινά ο αγώνας, η οποία μπορεί να πάρει τιμές από 0 έως και 22

Παράδειγμα: Ας υποθέσουμε ότι τα στοιχεία του εισιτηρίου είναι πύλη 18, σειρά G, θέση 72 για τις 27/11 ώρα 20:00.

Στο δυαδικό σύστημα, το 18 είναι 010010, η απόσταση του χαρακτήρα 'G' από τον 'A' είναι 00110, το 72 είναι 1001000, το 11 είναι 1011, το 27 είναι 11011 και το 20 είναι 10100. Επομένως, η είσοδος θα πρέπει να έχει την τιμή

0100 1000 1101 0010 0010 1111 0111 0100 η οποία στο δεκαεξαδικό σύστημα είναι 0x48D22F74

Το πρόγραμμά σας θα πρέπει να λειτουργεί ως εξής:

1. Δηλώνει μια μεταβλητή τύπου unsigned int .
2. Εκτυπώνει το μήνυμα "Enter code: " (με ένα space μετά την άνω-κάτω τελεία) και διαβάζει μια δεκαεξαδική τιμή την οποία αποθηκεύει σε αυτή τη μεταβλητή.
3. Ξεκινώντας από το λιγότερο σημαντικό bit της τιμής που διάβασε, χρησιμοποιεί κατάλληλους bitwise τελεστές για να αποσπάσει και να αποθηκεύσει σε ξεχωριστές μεταβλητές τις επιμέρους πληροφορίες.
4. Εκτυπώνει **χαρακτήρα αλλαγής γραμμής** και το μήνυμα "DD/MM, TT:00, G RS" όπου DD και MM, TT η ημέρα, μήνας και ώρα αντίστοιχα με πλάτος 2 χαρακτήρων και μηδενικό στην αρχή αν είναι μονοψήφια, G η πύλη, R η σειρά, S η θέση.
5. Οι θέσεις 1-40 είναι στο τμήμα A κάθε πύλης, οι θέσεις 41-80 στο τμήμα B και οι θέσεις 81-120 στο τμήμα C. Το πρόγραμμα ελέγχει τη θέση και για κάθε μία από τις παραπάνω περιπτώσεις εκτυπώνει **χαρακτήρα αλλαγής γραμμής**, το μήνυμα "Section X" όπου X είναι A ή B ή C και **χαρακτήρα αλλαγής γραμμής**.
6. Εκτυπώνει **χαρακτήρα αλλαγής γραμμής** και το μήνυμα "Enter today's time: " (με ένα space μετά την άνω-κάτω τελεία) και διαβάζει τη σημερινή ημερομηνία και ώρα η οποία θα δοθεί στη μορφή Η/Μ, Ω όπου Η η ημέρα, Μ ο μήνας και Ω η ώρα.
7. Εάν η ημερομηνία και ώρα του αγώνα είναι προγενέστερη αυτής που διαβάστηκε στο βήμα 6, εκτυπώνει **χαρακτήρα αλλαγής γραμμής** και το μήνυμα "MISSED GAME".
8. Εκτυπώνει **χαρακτήρα αλλαγής γραμμής**.