

ΕΡΓΑΣΤΗΡΙΟ 5

ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥΙ, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2009-2010

Στόχοι

- Διδιάστατοι πίνακες
- Συναρτήσεις
- Strings
- Εισαγωγή σε δείκτες

Πριν ξεκινήσετε

Αν δεν έχετε ήδη δημιουργήσει ένα φάκελο `ce120` μέσα στον προσωπικό σας χώρο, δημιουργήστε έναν τώρα.

Μέσα στο `ce120` δημιουργήστε ένα φάκελο `lab5`.

Δημιουργήστε ένα φάκελο μέσα στο `lab5` του οποίου το όνομα είναι

`επώνυμο_όνομα_AEM_email`

Για παράδειγμα, `doufexi_vana_000_doufexi@gmail.com`

Μέσα σε αυτό το φάκελλο θα βάζετε τις απαντήσεις σας πριν μας τις στείλετε σύμφωνα με τις οδηγίες σε επόμενη παράγραφο.

Ακολουθεί η άσκηση. Διαβάστε την εκφώνηση στο σύνολό της πριν αρχίσετε να λύνετε την άσκηση.

Μη διστάζετε να ζητήσετε βοήθεια!

Λυμένη άσκηση

Αν θέλουμε να αποθηκεύσουμε συμβολοσειρές σε ένα πίνακα, τότε ένας τρόπος είναι να χρησιμοποιήσουμε ένα διδιάστατο πίνακα χαρακτήρων. Γράψτε μια συνάρτηση `words` η οποία παίρνει ως παράμετρο ένα διδιάστατο πίνακα χαρακτήρων 2×10 , και αποθηκεύει σε αυτόν δύο λέξεις που διαβάζει από το πληκτρολόγιο.

Γράψτε μια συνάρτηση `main` η οποία

- δηλώνει ένα διδιάστατο πίνακα χαρακτήρων μεγέθους 2×10
- χρησιμοποιεί την `word` για να αποθηκεύσει δυο λέξεις σε αυτόν
- εκτυπώνει τα περιεχόμενα του πίνακα.

Χρησιμοποιείστε ανακατεύθυνση για να διαβάσετε και να γράψετε τα αποτελέσματα από και προς αρχεία αντί για το `standard input` και `output` αντίστοιχα.

Άλυτη άσκηση

Γράψτε μια συνάρτηση `read_words` η οποία παίρνει ως παράμετρο ένα διδιάστατο πίνακα χαρακτήρων 75×10 και αποθηκεύει σε αυτόν 75 λέξεις που διαβάζει από το πληκτρολόγιο (ή από αρχείο με ανακατεύθυνση). Μπορείτε να θεωρήσετε πως οι λέξεις έχουν το πολύ 10 χαρακτήρες μήκος, συμπεριλαμβανομένου του χαρακτήρα `'\0'`

Γράψτε μια συνάρτηση `find_longest` η οποία παίρνει ως παράμετρο ένα διδιάστατο πίνακα χαρακτήρων 75×10 στον οποίο έχουν ήδη αποθηκευτεί 75 λέξεις και βρίσκει και εκτυπώνει τη λέξη η οποία είναι λεξικογραφικά μεγαλύτερη. Μπορείτε να χρησιμοποιήσετε συναρτήσεις συμβολοσειρών από τη [βιβλιοθήκη της C](#).

Γράψτε μια συνάρτηση `main` η οποία χρησιμοποιεί τις συναρτήσεις `read_words` και `find_longest` για να διαβάσει 75 λέξεις και να βρει αυτή που είναι λεξικογραφικά μεγαλύτερη.

Αποθηκεύστε το πρόγραμμά σας σε ένα αρχείο με όνομα `lab5.c`.

Χρησιμοποιήστε το αρχείο `input.1` για να ελέγξετε την ορθότητα του προγράμματός σας. Η έξοδος του προγράμματός σας πρέπει να είναι η λέξη `why`.

Το πρόγραμμά σας πρέπει να περιέχει σχόλια συναρτήσεων ακριβώς όπως περιγράφεται στο φυλλάδιο με τίτλο [Πρότυπα σχολιασμού προγραμμάτων](#)

Αρχεία προς παράδοση: lab5.c

Πώς να παραδώσετε τη δουλειά σας

(Ακολουθείστε τις οδηγίες ακριβώς αλλιώς μπορεί να μη δούμε τα αρχεία σας)

Αν δεν είναι ήδη εκεί, μεταφέρετε τα παραδοτέα αρχεία από κάθε άσκηση στο φάκελο που φτιάξατε με όνομα `επώνυμο_όνομα_AEM_email`.

Πηγαίνετε στο φάκελο ο οποίος περιέχει το φάκελο `επώνυμο_όνομα_AEM_email` και γράψτε την παρακάτω εντολή:

```
tar czf επwnumo_onoma_AEM_email.tgz επwnumo_onoma_AEM_email
```

Στείλτε email στη διεύθυνση **ce120lab@gmail.com** και ένα αντίγραφο (CC) στον εαυτό σας με θέμα (subject) "CE120 lab5 tmimaX " (όπου X είναι ο αριθμός του τμήματος σας, δηλαδή 1-5) και επικολλημένο αρχείο το `επwnumo_onoma_AEM_email.tgz`

Υποβολές οι οποίες δεν ακολουθούν επακριβώς τις οδηγίες δημιουργίας των αρχείων και υποβολής δε θα λαμβάνονται υπόψη.

Για όσους τελειώσουν γρήγορα:

- Διαβάστε για τη συνάρτηση `sprintf` και χρησιμοποιήστε τη για να κατασκευάσετε κατάλληλο format string για μια `scanf` που διαβάζει μια συμβολοσειρά μεγέθους `SIZE`.
- Κατεβάστε το συμπιεσμένο αρχείο `pointerfun.tgz`, τρέξτε τα προγράμματα που περιέχει και πειραματιστείτε με pointers. Τα προγράμματα αυτά βασίζονται σε παραδείγματα που έχουν παρουσιαστεί στην τάξη.
- Κατεβάστε το συμπιεσμένο αρχείο `debug.tgz` και χρησιμοποιήστε `gdb` για να βρείτε τα λάθη στα προγράμματα που περιέχει.