

lab9: Δείκτες

Προετοιμασία:

1. Πλοηγηθείτε στον κατάλογο lab9 ο οποίος πρέπει να βρίσκεται μέσα στον κατάλογο ce120.
2. Ολοκληρώστε τις ασκήσεις της εβδομάδας. Τα αρχεία που θα γράψετε πρέπει να αποθηκευτούν στον κατάλογο lab9 που δημιουργήσατε προηγούμενα.
 - Θυμίζουμε πως η εντολή για να κάνετε compile ένα αρχείο με όνομα **lab9a.c** είναι:
gcc -Wall -g lab9a.c -o lab9a
Αν δεν υπάρχουν λάθη, θα δημιουργηθεί το εκτελέσιμο αρχείο **lab9a** το οποίο μπορείτε να εκτελέσετε γράφοντας **./lab9a**

Θέματα εργασίας:

- Δείκτες
- Συναρτήσεις
- Συμβολοσειρές
- Πίνακες
- Χρήση εντολών ελέγχου και επανάληψης
- Αναγνωσιμότητα κώδικα (σχόλια, στοίχιση, περιγραφικά ονόματα μεταβλητών, κτλ.)

Ελεγχος ορθότητας:

Για κάθε άσκηση θα σας δώσουμε ενδεικτικά αρχεία εισόδου και εξόδου. Πέρα από τους ελέγχους που θα κάνετε εσείς, συνιστούμε να τρέξετε τα προγράμματά σας και με τις δικές μας εισόδους, να αποθηκεύσετε τις αντίστοιχες εξόδους σας σε αρχεία με χρήση ανακατεύθυνσης και μετά να τις συγκρίνετε με τις δικές μας εξόδους με χρήση της εντολής diff.

Για παράδειγμα, αν σας έχουμε δώσει το αρχείο εισόδου in1.txt και την αντίστοιχη έξοδο std1.txt, τρέξτε

```
./lab9a < in1.txt > out1.txt
```

για να αποθηκεύσετε την έξοδό σας στο αρχείο out1.txt, και μετά συγκρίνετε

```
diff out1.txt std1.txt
```

Αν δεν υπάρχουν διαφορές, δε θα εμφανιστεί τίποτα. Αν υπάρχουν, θα εμφανιστούν ανά γραμμή

Ασκηση 1

Το πρόγραμμα που θα γράψετε για την άσκηση πρέπει να το αποθηκεύσετε σε αρχείο με όνομα **lab9.c** μέσα στον κατάλογο lab9.

Γράψτε μια συνάρτηση (1η συνάρτηση) η οποία λαμβάνει ως όρισμα ένα μονοδιάστατο πίνακα χαρακτήρων και το μέγεθος του πίνακα και έχει τύπο επιστροφής **void**. Η συνάρτηση κάνει τα εξής:

- Αρχικοποιεί τον πίνακα που λαμβάνει ως όρισμα ώστε να περιέχει σε όλες τις θέσεις του την τιμή **'\0'**.
- Σε μία επανάληψη κάνει τα εξής:
 - Διαβάζει μία συμβολοσειρά και την αποθηκεύει προσωρινά σε ένα μονοδιάστατο πίνακα μεγέθους **SIZE**.
 - Εάν η συμβολοσειρά που διαβάστηκε δεν χωράει στον πίνακα ή διαβαστεί η συμβολοσειρά **"##quit##"** η συνάρτηση επιστρέφει.
 - Εάν η συμβολοσειρά χωράει στον πίνακα που λαμβάνει ως όρισμα την αντιγράφει σε αυτόν, αμέσως μετά την τελευταία συμβολοσειρά που προστέθηκε στον πίνακα (υποχρεωτικά με χρήση **strcpy**). **Προσοχή:** Στον πίνακα οι συμβολοσειρές θα πρέπει να παραμένουν διακριτές μεταξύ τους, δηλαδή δεν πρέπει να ενώνονται σε μία.

Γράψτε μια συνάρτηση (2η συνάρτηση) η οποία λαμβάνει ως όρισμα ένα μονοδιάστατο πίνακα χαρακτήρων, το μέγεθος του πίνακα χαρακτήρων, ένα μονοδιάστατο πίνακα δεικτών σε χαρακτήρα και το μέγεθος του πίνακα δεικτών και κάνει τα εξής:

- Αρχικοποιεί τον πίνακα δεικτών ώστε να περιέχει την τιμή **NULL** σε όλες τις θέσεις του.
- Ψάχνει στον πίνακα χαρακτήρων όλες τις αρχικές θέσεις των συμβολοσειρών που περιέχει αυτός (**hint:** όλες τις θέσεις που δεν περιέχουν την τιμή **'\0'**, αλλά έπονται της τιμής **'\0'**. Εξαίρεση αποτελεί η διεύθυνση της 1ης θέσης του πίνακα χαρακτήρων). Κάθε τέτοια θέση που βρίσκει, αποθηκεύει τη διεύθυνσή της στον πίνακα δεικτών (σε διαδοχικές θέσεις του πίνακα δεικτών). Οι θέσεις των συμβολοσειρών θα πρέπει να διατηρούν τη σειρά που έχουν στον πίνακα χαρακτήρων.

Η συνάρτηση έχει τύπο επιστροφής **void**.

Γράψτε μια συνάρτηση (3η συνάρτηση) η οποία λαμβάνει ως όρισμα ένα μονοδιάστατο πίνακα δεικτών σε χαρακτήρα και το μέγεθος του και εκτυπώνει στην οθόνη όλες τις συμβολοσειρές στις οποίες δείχνουν οι δείκτες που περιέχονται στον πίνακα δεικτών, χωρισμένες μεταξύ τους με χαρακτήρα αλλαγής γραμμής. Η συνάρτηση έχει τύπο επιστροφής **void**.

Γράψτε τη συνάρτηση main η οποία ορίζει ένα πίνακα χαρακτήρων μεγέθους **SIZE** και ένα πίνακα δεικτών μεγέθους **SIZE/2**. Στη συνέχεια καλεί την 1η συνάρτηση για να γεμίσει τον πίνακα χαρακτήρων, καλεί τη 2η συνάρτηση για να γεμίσει τον πίνακα δεικτών και στο τέλος καλεί την 3η συνάρτηση για να εκτυπώσει τα περιεχόμενα του πίνακα χαρακτήρων στην οθόνη με τη βοήθεια του πίνακα δεικτών.

Χρησιμοποιήστε **SIZE=64**

Προσοχή: Πρέπει να χρησιμοποιούνται συναρτήσεις από το **string.h** όπου είναι δυνατό.

Αρχεία εισόδου: **in1.txt**, **in2.txt**, **in3.txt**

Αντίστοιχα αρχεία εξόδου: **std1.txt**, **std2.txt**, **std3.txt**

Άσκηση 2 (προαιρετική, αλλά αν δεν προλάβετε στο εργαστήριο, ολοκληρώστε τη στο σπίτι).

Για αυτή την άσκηση θα γράψετε ένα πρόγραμμα το οποίο διαβάζει δύο συμβολοσειρές (ας τις πούμε `original` και `delimiter`) κι ανιχνεύει αν η `delimiter` εμφανίζεται τουλάχιστον δύο φορές στην `original`. Αν ναι, θεωρούμε ότι η `delimiter` οριοθετεί τρία τμήματα της `original`:

- τμήμα 1: πριν την πρώτη εμφάνιση της `delimiter` στην `original`,
- τμήμα 2: ανάμεσα στις δύο πρώτες εμφανίσεις της `delimiter` στην `original`,
- τμήμα 3: μετά τη δεύτερη εμφάνιση της `delimiter` στην `original`,

Στόχος μας είναι να κατασκευάσουμε μια τρίτη συμβολοσειρά η οποία θα αποτελείται από το τμήμα 2 της `original` χρησιμοποιώντας δείκτες όπου είναι δυνατό. Όλες οι συμβολοσειρές θα είναι αποθηκευμένες σε πίνακες μεγέθους `SIZE`.

Υλοποιήστε μια συνάρτηση `getPart` η οποία:

1. Παίρνει ως παραμέτρους τρεις πίνακες χαρακτήρων. Οι δύο πρώτοι περιέχουν τις συμβολοσειρές που ονομάσαμε `original` και `delimiter` πιο πάνω και ο τρίτος είναι αρχικά άδειος.
2. Η συνάρτηση εντοπίζει τα τρία τμήματα της `original` όπως αυτά οριοθετούνται από τη `delimiter` και αντιγράφει το μεσαίο τμήμα στον πίνακα που περάστηκε ως τρίτη παράμετρος.
3. Η `getPart` επιστρέφει ακέραιο κωδικό λάθους, `OK` αν όλα πήγαν καλά και αποθηκεύτηκε συμβολοσειρά στην τρίτη παράμετρο ή `NOT_FOUND` αν δε βρέθηκαν τουλάχιστον δύο εμφανίσεις της `delimiter` στην `original`.

Υλοποιήστε μια συνάρτηση `main` η οποία:

1. Εκτυπώνει το μήνυμα **Enter string:**
2. Διαβάζει την αρχική συμβολοσειρά (αυτή που ονομάσαμε `original`)
3. Εκτυπώνει το μήνυμα **Enter delimiter:**
4. Διαβάζει τη συμβολοσειρά που οριοθετεί τα τμήματα της `original` (αυτή που ονομάσαμε `delimiter`)
5. Καλεί τη συνάρτηση `getPart` η οποία παίρνει ως παραμέτρους τις παραπάνω συμβολοσειρές καθώς ένα ακόμη πίνακα χαρακτήρων, αρχικά άδειο.
 - Αν η `getPart` επιστρέψει `OK`, εκτυπώνεται **η νέα συμβολοσειρά** που δημιουργήθηκε, ακολουθούμενη από **χαρακτήρα αλλαγής γραμμής**.
 - Αν η συνάρτηση επιστρέψει `NOT_FOUND` εκτυπώνεται το μήνυμα **X does not appear twice in Y** όπου `X` η τιμή της `delimiter` και `Y` η τιμή της `original`. Ακολουθεί **χαρακτήρας αλλαγής γραμμής**.

ΥΠΟΘΕΣΕΙΣ/ΑΠΑΙΤΗΣΕΙΣ:

- Χρησιμοποιήστε `SIZE 32`, αλλά το πρόγραμμα πρέπει να είναι γραμμένο έτσι ώστε να λειτουργήσει σωστά ακόμη κι αν το `SIZE` αλλάξει.
- Όλες οι μεταφορές και προσπελάσεις συμβολοσειρών πρέπει να γίνουν αποκλειστικά με χρήση δεικτών και πράξεων σε δείκτες. ΑΠΑΓΟΡΕΥΕΤΑΙ ΑΥΣΤΗΡΑ να χρησιμοποιηθεί σύνταξη πινάκων (αγκύλες) για οτιδήποτε άλλο εκτός από δηλώσεις πινάκων στη `main` και στη λίστα παραμέτρων της συνάρτησης.
- Προσοχή: Συναρτήσεις όπως `strcpy`, `strcat` έχουν απροσδιόριστη συμπεριφορά όταν οι παράμετροι είναι τμήματα της ίδιας συμβολοσειράς και δεν πρέπει να χρησιμοποιηθούν.

Αποστολή Προγραμμάτων για σχολιασμό και βαθμολόγηση

- Πλοηγηθείτε στον κατάλογο **lab9**. Θα πρέπει να βρίσκεται μέσα σε αυτόν το αρχείο **lab9.c**
- Κατασκευάστε να κατάλογο με όνομα lab9_επώνυμο1_AEM1_επώνυμο2_AEM2 (ΜΕ ΛΑΤΙΝΙΚΟΥΣ ΧΑΡΑΚΤΗΡΕΣ) όπου επώνυμο1, AEM1 αναφέρονται στο ένα μέλος της ομάδας και επώνυμο2, AEM2 στο δεύτερο μέλος. Για παράδειγμα, αν τα δύο μέλη της ομάδας ήταν οι Ντουφεξή με AEM 01234 και Θάνος με AEM 05678, θα γράφατε:
mkdir lab9_Doufexi_01234_Thanos_05678
- Αντιγράψτε το αρχείο lab9.c μέσα στον κατάλογο lab9_επώνυμο1_AEM1_επώνυμο2_AEM2
- Πακετάρετε τον κατάλογο (compress here as tar.gz).
- Το ένα μέλος της ομάδας:
 - μπαίνει στο λογαριασμό του στο eclass,
 - πηγαίνει στην ενότητα Εργασίες,
 - κάνει click στην εργασία lab9 για το συγκεκριμένο τμήμα
 - **ΠΡΟΣΟΧΗ!!!!** Στο πεδίο Ομάδα Χρηστών επιλέγει τη σωστή ομάδα (με βάση τα ονόματα των μελών)
 - Ανεβάζει το .tar.gz αρχείο μέσω του Browse **και πατά Αποθήκευση.**
 - **ΠΡΟΣΟΧΗ!!!!** Επιβεβαιώνει ότι ανέβασε το σωστό αρχείο.