

lab9: Δείκτες

Πριν ξεκινήσετε...

Πατήστε πάνω στο σύνδεσμο "Download handout" και κατεβάστε το αρχείο lab9.tar.gz στον υπολογιστή σας. Στη συνέχεια ανοίξτε το αρχείο lab9.tar.gz με το ποντίκι και σύρετε (drag & drop) τον κατάλογο lab9 μέσα στο φάκελο ce120 που βρίσκεται στο home directory σας. Θα δουλέψετε μέσα στον κατάλογο lab9 που μόλις μεταφέρατε.

Διαχωριστικό string

Οπου σας λέμε να εκτυπώσετε το **διαχωριστικό string**, εισάγετε μια `printf("\n#\n");`;

Υποβολή ασκήσεων

1. **Πριν παραδώσετε τις ασκήσεις βεβαιωθείτε ότι είναι σωστή η ομάδα σας στο Autolab κι αν δεν είναι, φτιάξτε τη ξανά!**
2. **Αντιγράψτε** τα `lab9a.c`, `lab9b.c` στον κατάλογο `lab9submit` που βρίσκεται μέσα στο `lab9`.
 - Δεν πρέπει να υπάρχουν άλλα αρχεία σε αυτόν τον κατάλογο.
3. Κάντε **δεξί κλικ** στον κατάλογο `lab9submit` και επιλέξτε `Compress` → `Here as tar.gz`
4. Θα έχει δημιουργηθεί ένα αρχείο με όνομα `lab9submit.tar.gz` το οποίο ένα μέλος της ομάδας (δεν έχει σημασία ποιο) **υποβάλλει** στο Autolab.

Προσοχή:

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

* Για ονόματα που ορίζονται με `#define`, το πρόγραμμα πρέπει να είναι γραμμένο ώστε αν αλλαχθεί η τιμή του ονόματος, τότε το πρόγραμμα να συνεχίσει να λειτουργεί σωστά.

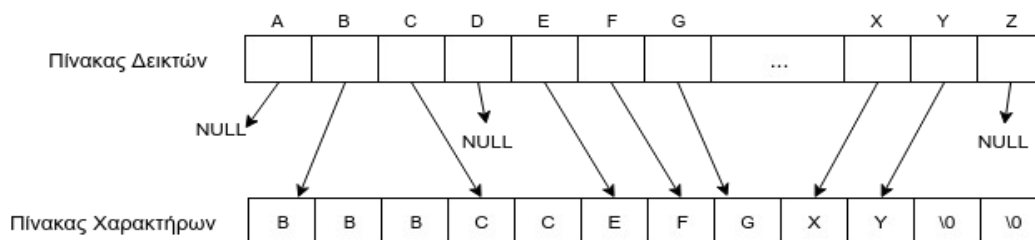
Άσκηση 1

Προσθέστε τον κώδικά σας στο έτοιμο αρχείο με όνομα **lab9a.c** που βρίσκεται στον κατάλογο lab9 που προέκυψε από την αποσυμπίεση του lab9.tar.gz.

Διαβάστε όλη την εκφώνηση και δείτε τα περιεχόμενα του lab9a.c πριν ξεκινήσετε.

Το πρόγραμμά σας θα πρέπει να κάνει τα εξής:

1. Αρχικοποιεί ένα πίνακα δεικτών σε χαρακτήρα μεγέθους 26 θέσεων και ένα πίνακα χαρακτήρων μεγέθους MAX_SIZE (ορίζεται στο lab9.h). Ο πίνακας δεικτών έχει τόσες θέσεις όσα είναι και τα γράμματα του αγγλικού αλφαβήτου.
2. Παίρνει ως όρισμα προγράμματος από τη γραμμή εντολών έναν ακέραιο αριθμό με τιμές μεταξύ 1 και MAX_TEST (συμπεριλαμβανομένων). Εάν πάρει διαφορετική τιμή το πρόγραμμα εκτυπώνει **"Invalid input"** και **χαρακτήρα αλλαγής γραμμής** και τερματίζει. Αν δε δοθεί όρισμα, το πρόγραμμα εκτυπώνει **"Wrong arguments"** και **χαρακτήρα αλλαγής γραμμής** και τερματίζει.
3. Καλεί την έτοιμη συνάρτηση **init_array**, η οποία λαμβάνει ως ορίσματα ένα πίνακα χαρακτήρων και έναν ακέραιο. Κατά την κλήση της, το πρώτο όρισμα είναι ο πίνακας χαρακτήρων που δηλώσατε στο βήμα 1 και το δεύτερο όρισμα ο ακέραιος που διαβάσατε στο βήμα 2. Το δεύτερο όρισμα καθορίζει τι είδους δεδομένα θα αποθηκευτούν στον πίνακα. Χρησιμοποιείται για να μπορέσουμε να ελέγξουμε την ορθότητα του προγράμματος για διάφορες εισόδους, και δε χρειάζεται να ξέρετε κάτι άλλο γι αυτό. Μετά την κλήση της συνάρτησης ο πίνακας χαρακτήρων θα είναι ταξινομημένος και περιέχει μόνο κεφαλαία γράμματα του αγγλικού αλφαβήτου ή '\0' σε όσες θέσεις μένουν κενές στο τέλος του πίνακα.



4. Κάθε θέση του πίνακα δεικτών θα πρέπει να αντιστοιχεί στο αντίστοιχο γράμμα του αγγλικού αλφαβήτου. Για παράδειγμα η θέση 0 αντιστοιχεί στο 'A', η θέση 1 στο 'B', η θέση 2 στο 'C' κ.ο.κ. Αρχικοποιήστε τον πίνακα δεικτών ως εξής: εντοπίστε την 1η εμφάνιση κάθε γράμματος μέσα στον πίνακα χαρακτήρων και αποθηκεύστε στην αντίστοιχη θέση του πίνακα δεικτών τη διεύθυνση του γράμματος αυτού. Εάν ένα γράμμα δεν εμφανίζεται μέσα στον πίνακα χαρακτήρων τότε η διεύθυνση της αντίστοιχης θέσης του πίνακα δεικτών είναι NULL. Δείτε ενδεικτικά το παραπάνω σχήμα. Η θέση 0 του πίνακα δεικτών είναι NULL γιατί δεν υπάρχει χαρακτήρας 'A' στον πίνακα χαρακτήρων. Η θέση 1 του πίνακα δεικτών δείχνει στη θέση του πίνακα χαρακτήρων όπου βρίσκεται το πρώτο 'B', κ.ο.κ.
5. Συμπληρώστε την υλοποίηση της συνάρτησης **print** η οποία παίρνει ως παράμετρο τον πίνακα δεικτών και να εκτυπώνει με τη σειρά κάθε γράμμα του αγγλικού αλφαβήτου, άνω-κάτω τελεία, κενό και, αν το γράμμα υπάρχει, έναν ακέραιο που ισοδυναμεί με τη θέση της 1ης εμφάνισης του γράμματος αυτού στον πίνακα χαρακτήρων, διαφορετικά τη λέξη "NULL". Για το προηγούμενο παράδειγμα, η εκτύπωση θα είναι όπως στο σχήμα που ακολουθεί. Παρατηρήστε ότι υπάρχει ήδη κώδικας στη main που καλεί αυτή τη συνάρτηση.

A: NULL	N: NULL
B: 0	O: NULL
C: 3	P: NULL
D: NULL	Q: NULL
E: 5	R: NULL
F: 6	S: NULL
G: 7	T: NULL
H: NULL	U: NULL
I: NULL	V: NULL
J: NULL	W: NULL
K: NULL	X: 8
L: NULL	Y: 9
M: NULL	Z: NULL

--> Συνέχεια στη διπλανή στήλη

Άσκηση 2 (Προαιρετική)

Αποθηκεύστε τον κώδικά σας σε αρχείο με όνομα **lab9b.c** στον κατάλογο lab9 που προέκυψε από την αποσυμπίεση του lab9.tar.gz.

Η άσκηση θα πρέπει να λυθεί αποκλειστικά με χρήση συναρτήσεων από το string.h και όχι με χρήση επαναλήψεων.

Απαιτήσεις: Όλες οι συμβολοσειρές του προγράμματος πρέπει να αποθηκευτούν σε πίνακες μεγέθους MAX_SIZE. Ορίστε το MAX_SIZE ίσο με 10. Το πρόγραμμά σας θα πρέπει να είναι γραμμένο ώστε να λειτουργήσει σωστά ακόμη κι αν αυτή η τιμή αλλάξει. .

Το πρόγραμμα σας θα πρέπει να κάνει τα εξής:

1. Εκτυπώνει **χαρακτήρα αλλαγής γραμμής** και το μήνυμα **"Enter word: "** (με ένα space μετά το ':') και διαβάζει μια συμβολοσειρά.
2. Ελέγχει αν η συμβολοσειρά είναι της μορφής ss δηλαδή αποτελείται από δύο όμοιες συμβολοσειρές. Αν ναι, εκτυπώνει **χαρακτήρα αλλαγής γραμμής**, το μήνυμα **"S is a twin string"** όπου S η συμβολοσειρά που διαβάστηκε και **χαρακτήρα αλλαγής γραμμής**. Αν δεν είναι, εκτυπώνει **χαρακτήρα αλλαγής γραμμής**, το μήνυμα **"S is not a twin string"** όπου S η συμβολοσειρά που διαβάστηκε και **χαρακτήρα αλλαγής γραμμής**.
3. Εκτυπώνει το **διαχωριστικό** string.
4. Επαναλαμβάνει τα βήματα 1-3 έως ότου δοθεί ως είσοδος η συμβολοσειρά **"quit"**

Για παράδειγμα, οι συμβολοσειρές "abcabc", "413_413_", "xx" είναι twin ενώ οι "a", "abcabC", "123-123", "123321" δεν είναι.