

## Παρατηρήσεις πάνω στο lab9

### Ορίσματα προγράμματος

Τα ορίσματα ενός προγράμματος αποθηκεύονται στον πίνακα `argv`. Στη θέση μηδέν βρίσκεται το όνομα του εκτελέσιμου και από τις θέσεις 1 και πέρα τα υπόλοιπα ορίσματα ως strings. Πριν προσπελάσουμε οποιοδήποτε από αυτά, πρέπει ΠΑΝΤΑ να ελέγχουμε την τιμή του `argc` η οποία μας λέει πόσα ορίσματα υπάρχουν. Στη συγκεκριμένη άσκηση θέλουμε ένα πλεόν του εκτελέσιμου, άρα ελέγχουμε αν το `argc` είναι 2 και μόνο τότε προχωράμε.

Εφόσον τα ορίσματα αποθηκεύονται ως strings, αν κάποιο από τα ορίσματα έχει αριθμητική τιμή θα πρέπει να το μετατρέψουμε. Για τη μετατροπή σε ακέραιο, που καλούσε η άσκηση, χρησιμοποιούμε τη συνάρτηση `atoi` για την οποία χρειάζεται `#include<stdlib.h>`. Η σύνταξη είναι `atoi(argv[1])`. Κατά κανόνα είναι λάθος να γίνει η μετατροπή με αφαίρεση του `'0'` από τον πρώτο χαρακτήρα του `argv[1]` (δηλαδή `*argv[1] - '0'`) γιατί κάτι τέτοιο κάνει σωστή μετατροπή μόνο το `argv[1]` αντιστοιχεί σε μονοψήφιο ακέραιο.

### Αρχικοποιήσεις

Είναι καλή ιδέα να αρχικοποιείτε πάντα τις συμβολοσειρές σας ώστε να περιέχουν παντού `'\0'` και τους δείκτες ώστε να είναι NULL. Ο πιο απλός τρόπος να αρχικοποιηθούν όλα τα στοιχεία ενός πίνακα σε μηδενική τιμή είναι κατά τη δήλωση του πίνακα, για παράδειγμα:

```
char letters[SIZE] = {'\0'};
char *ptrs[SIZE] = {NULL};
int numbers[SIZE] = {0};
```

Όταν αρχικοποιούμε ένα πίνακα με τέτοιο τρόπο, τότε το πρώτο στοιχείο του παίρνει την τιμή που έχουμε ανάμεσα στα άγκιστρα, και όλα τα υπόλοιπα γίνονται μηδέν. Άρα στα παραπάνω παραδείγματα θα γίνουν μηδέν και τα SIZE στοιχεία του κάθε πίνακα. Ενώ αν είχαμε `int numbers[SIZE] = {15};` τότε θα γινόταν το πρώτο 15 και μετά όλα τα υπόλοιπα μηδέν.

### Βήμα 4

Στη συζήτηση που ακολουθεί, αναφερόμαστε στον πίνακα δεικτών ως `ptrs` και στον πίνακα χαρακτήρων ως `letters`.

Στο βήμα 4 στόχος είναι η κατασκευή του `ptrs`. Όπως πάντα, προσέχουμε να αρχικοποιήσουμε σωστά τον πίνακα από την αρχή ώστε όλα τα κελιά του να περιέχουν NULL:

```
char *ptrs[26] = {NULL};
```

Για να βρούμε τα μη-κενά κελιά, πρέπει να διατρέξουμε τον πίνακα `letters` και κάθε φορά που βρίσκουμε καινούργιο γράμμα (δηλαδή διαφορετικό από το προηγούμενο), να θέτουμε τον αντίστοιχο δείκτη. Προσοχή: αν σκοπεύουμε να προσπελάσουμε το γράμμα που είναι μια θέση πριν την τρέχουσα, πρέπει να γράψουμε προσεκτικά τον έλεγχο του πρώτου γράμματος για να μη βγούμε εκτός ορίων πίνακα. Για το τέλος της επανάληψης ελέγχουμε το μέγεθος του πίνακα. Μπορούμε να ελέγξουμε κι αν δεν υπάρχουν άλλα γράμματα (έλεγχος για `'\0'`) ώστε να τερματίσουμε πρόωρα, αλλά δεν είναι απαραίτητο. Τέλος, εφόσον θέλουμε ο αύξων αριθμός κάθε μη-κενού κελιού του `ptrs` να είναι όσο η απόσταση του αντίστοιχου γράμματος από την αρχή της αλφαβήτας, θα χρησιμοποιήσουμε αριθμητική χαρακτήρων. Μια ενδεικτική λύση είναι:

```
int pos=0;
for (pos = 0; pos < MAX_SIZE && letters[pos] != '\0'; pos++) {
    if (pos == 0 || letters[pos] != letters[pos-1]) {
        ptrs[letters[pos] - 'A'] = &letters[pos];
    }
}
```

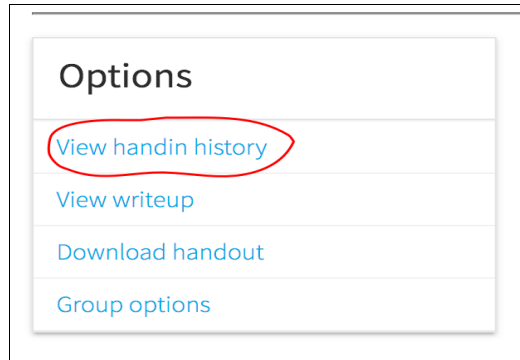
Σε αυτό το σημείο, θα μπορούσατε να προσθέσετε προσωρινά μια `printf` που εκτυπώνει το `pos` και το `*ptrs[letters[pos] - 'A']` ώστε να βεβαιωθείτε ότι γίνεται η ανάθεση που πρέπει, πριν προχωρήσετε στο επόμενο βήμα.

Εξίσου σωστή είναι η χρήση διπλής for όπως κάνατε αρκετοί, αλλά σε αυτή την περίπτωση πρέπει να προσέξετε πρώτον το εξωτερικό loop να είναι αυτό που διατρέχει τα γράμματα της αλφαβήτας και δεύτερον να κάνετε break αμέσως μόλις βρείτε το πρώτο γράμμα κάθε σειράς, διαφορετικά ο pointer θα μετακινηθεί στα επόμενα όμοια γράμματα:

```
int i, pos=0;
for (i=0; i < 26; i++) {
    for (pos = 0; pos < MAX_SIZE && letters[pos] != '\0'; pos++) {
        if (letters[pos] == 'A'+i) {
            ptrs[i] = &letters[pos];
            break;
        }
    }
}
```

## Πώς να βλέπετε τα σχόλια βαθμολόγησης

Για να δείτε τα σχόλια βαθμολόγησης, κάντε login στο autolab, πηγαίνατε στο lab που σας ενδιαφέρει και επιλέξτε View Handin History:



Κάτω από κάθε κατηγορία βαθμολόγησης θα δείτε το βαθμό σας σε αυτή. Κάνετε κλικ πάνω στο βαθμό για να δείτε συγκεκριμένες παρατηρήσεις. **Στη στήλη PARATHRHSEIS έχετε όλοι σχόλια για το πρόγραμμά σας. Διαβάστε τα!**

	Submission (0.0)	a_compilation (0.0)	a_tests (21.0)	a_results (24.0)	a_style (30.0)	b_compilation (0.0)	b_tests (16.0)	b_style (9.0)	PARATHRHSEIS (0.0)
	0.0	-15.0	9.0	24.0	21.0	-5.0	16.0	9.0	0.0

## Αντιστοιχία βαθμού lab9 και πόντων εργαστηρίου

Βαθμός lab8	Πόντοι εργαστηρίου
0-39	0
40-49	1
50-100	3