

Παρατηρήσεις πάνω στο lab8

Ονόματα μεταβλητών και στοίχιση

Τώρα που τα προγράμματα είναι πιο σύνθετα, η σωστή στοίχιση και τα περιγραφικά ονόματα μεταβλητών είναι σημαντικά στοιχεία του κώδικα που βοηθούν στην κατανόησή του και στην εύρεση λαθών. Ο λόγος που επιμένουμε σε αυτό δεν είναι για έχουμε ένα παραπάνω πράγμα να διορθώσουμε αλλά γιατί πραγματικά σας βοηθά. Γι αυτό το λόγο είναι σημαντικό να έχετε από την αρχή καλά ονόματα και στοίχιση αντί να λέτε "ας το κάνω πρόχειρα τώρα και το αλλάζω μετά." Μετά, όταν για παράδειγμα θα έχετε μπερδέψει στο μυαλό σας τι είναι το `str1`, τι το `str2` και τι το `str3`, θα είναι πολύ δύσκολο να βρείτε πού έχετε κάνει λάθος και αν αυτό που θέλετε είναι `strcat(str1, str2)` ή `strcat(str2, str1)`.

Αλγόριθμος

Η άσκηση ζητούσε να διαβάζονται λέξεις κι όσο είναι δυνατό να επικολλούνται σε ένα μεγάλο string μαζί με ένα space κάθε φορά. Έχουμε δύο συνθήκες να ελέγξουμε: (α) αν η λέξη είναι ίδια με την προηγούμενη, τότε την παρακάμπτουμε (β) αν η λέξη μαζί με το space δε θα χωρέσουν, τότε σταματάμε.

Μια ενδεικτική (και όχι μοναδική) σειρά των πράξεων είναι:

```
while (1) {
    διάβασε μια λέξη
    αν είναι ίδια με την προηγούμενη,
        continue
    αν η λέξη μαζί με το space και μαζί με το '\0' θα χωρέσουν στο μεγάλο string
        strcat το space
        strcat τη λέξη
        αντίγραψε τη λέξη πάνω στην προηγούμενη
    διαφορετικά
        break
}
```

Παρατήρηση 1

Για να χωρέσουν η λέξη μαζί με το space στο μεγάλο string θα πρέπει το `MAX_SIZE` μείον το τρέχον μέγεθος του μεγάλου string να είναι τουλάχιστον όσο το μήκος της λέξης συν ένα για το `'\0'` συν ένα για το space. Μην ξεχνάτε ότι η `strlen` δεν υπολογίζει το `'\0'`.

Παρατήρηση 2

Οι συναρτήσεις για strings βασίζονται στο ότι το τέλος ενός string σηματοδοτείται από το `'\0'`. Για παράδειγμα, η `strcat(x, y)` θα ψάξει πρώτα στο `x` να βρεί το `'\0'` και μετά θα αρχίσει να αντιγράφει το `y` στο `x` από εκείνο το σημείο και μετά. Αυτό σημαίνει ότι αν το `x` δεν έχει `'\0'` εκεί που πρέπει, η `strcat` δε θα λειτουργήσει σωστά.

Η πρώτη `strcat` που εκτελείται στην άσκηση είναι αυτή που θα κολήσει το space στην αρχή του μεγάλου string. Για να γίνει σωστά αυτό, θα πρέπει το μεγάλο string να έχει `'\0'` στην αρχή του, δηλαδή: `sentence[0] = '\0'`; Αυτό που προτείνουμε όμως είναι (α) να αρχικοποιείτε πάντα όλες τις θέσεις ενός string με `'\0'` και (β) να έχετε πάντα στην άκρη του μυαλού σας τη σκέψη ότι το string σας πρέπει καθόλη τη διάρκεια του προγράμματος να έχει `'\0'` στη σωστή θέση. Η αρχικοποίηση μπορεί να γίνει εύκολα κατά τη δήλωση: `char sentence[MAX_SIZE] = {'\0'};`

Παρατήρηση 3

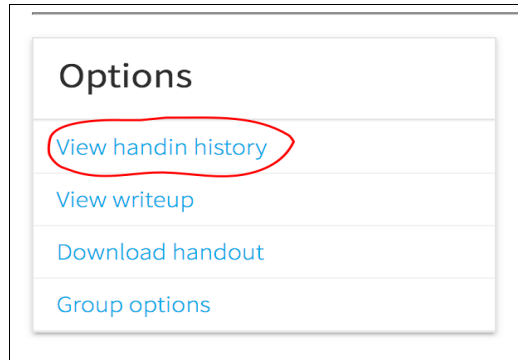
Ένα συχνό και πολύ σοβαρό λάθος ήταν να γίνεται πρώτα η `strcat` και μετά ο έλεγχος για το αν η λέξη χωράει. Το αποτέλεσμα σε αυτή την περίπτωση είναι ότι αν η λέξη δε χωράει, θα έχει ήδη εισαχθεί και θα έχουμε υπερβεί τα όρια του πίνακα.

Παρατήρηση 4

Εφόσον το πλήθος των επαναλήψεων εξαρτάται από τα μήκη των λέξεων, δεν είναι γνωστό από πριν, άρα η καλύτερη επιλογή για τύπο loop είναι `while`.

Πώς να βλέπετε τα σχόλια βαθμολόγησης

Για να δείτε τα σχόλια βαθμολόγησης, κάντε login στο autolab, πηγαίνετε 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.

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

Στο lab8 έχουμε γράψει παρατηρήσεις στις στήλες style και PARATHRHSEIS. Αν κάνετε click σε αυτά θα δείτε για παράδειγμα,

```
Κακή στοίχιση: -10  
Score for this problem: -10.0
```

Αντιστοιχία βαθμού lab8 και πόντων εργαστηρίου

Βαθμός lab8	Πόντοι εργαστηρίου
0-25	0
26-45	1
46-100	3