

Παρατηρήσεις για το lab8

Μεταγλώττιση: Ο κώδικας θα πρέπει να μεταγλωττίζεται χωρίς **warnings**. Χαρακτηρισμός “μη ικανοποιητικό” σημαίνει ότι είχατε τουλάχιστον ένα warning. Κώδικας που παράγει λάθη (errors) κατά τη μεταγλώττιση βαθμολογείται με FAIL.

Στοιχισιη/Κενά: Ο κώδικας πρέπει να είναι στοιχισμένος σωστά, με ιδιαίτερη προσοχή στη στοιχισιη εμφωλευμένων for/while/if. Δεν πρέπει να έχει περισσότερες από μία διαδοχικές κενές γραμμές, αλλά ούτε και να είναι όλα "κολλημένα".

Χαρακτηρισμός “μέτρια” γενικά σημαίνει ότι έχετε κατά το πλείστον καλή στοιχισιη και χρήση κενών γραμμών. Χαρακτηρισμός “μη ικανοποιητική” σημαίνει ότι δεν έχετε στοιχισιη τον κώδικά σας ή η στοιχισιη είναι πολύ ασυνεπής ειδικά όσον αφορά τις δομές ελέγχου/επανάληψης.

Είσοδος: Στην πρώτη άσκηση δώστε ιδιαίτερη προσοχή στον έλεγχο ορθότητας της εισόδου (εύρος τιμής). Στη δεύτερη άσκηση το μέγεθος του πίνακα πρέπει να ορίζεται με **#define**, και πρέπει να χρησιμοποιείται το όνομα της σταθεράς σε όλο το πρόγραμμα. Για την είσοδο πρέπει να χρησιμοποιηθεί **sprintf(format, "%ds", SIZE-1)** για την κατασκευή κατάλληλου format string. Προσοχή στο -1. Δεν είναι σωστό να χρησιμοποιηθεί SIZE γιατί τότε το '\0' μπορεί να αποθηκευτεί μετά το τέλος του πίνακα. Αυτό μπορείτε να το ελέγξετε τρέχοντας το 4ο τεστ.

Μέτριο = Χρήση "%19s" αντί για παραμετροποιημένο format string
Μη ικανοποιητικό = λάθος στη συνθήκη (ασκ. 1), χρήση "%s" (ασκ.2)

Ορισμός και κλήσεις συναρτήσεων: Για κάθε συνάρτηση ελέγχουμε προσεκτικά το prototype (παράμετροι και τύποι αυτών, τύπος επιστροφής), αν καλείται σωστά και, στην περίπτωση που επιστρέφει κάτι, αν η τιμή επιστροφής αποθηκεύεται σε κατάλληλη μεταβλητή. Ο τύπος της μεταβλητής που περιέχει την τιμή επιστροφής πρέπει να είναι ίδιος με τον τύπο επιστροφής της συνάρτησης. Για παράδειγμα, εφόσον η συνάρτηση παραγοντικού επιστρέφει long int, θα πρέπει η τοπική μεταβλητή στην οποία αποθηκεύεται το αποτέλεσμα πριν επιστραφεί να είναι επίσης long int.

Ένα συχνό λάθος είναι να καλείται 2 φορές μια συνάρτηση, με το αποτέλεσμα της πρώτης κλήσης να μη χρησιμοποιείται. Για παράδειγμα:

```
factorial(n);  
x = factorial(n);
```

Τέλος, προσέξτε τη σειρά κλήσης των συναρτήσεων. Καλούμε την series_sinh από τη main, την series_sinh_factor από την series_sinh, την factorial από τη series_sinh_factor. Αντίστοιχα στην άσκηση 1 της Παρασκευής

Μέτριο = άσκοπες κλήσεις συναρτήσεων (όπως η factorial πιο πάνω)

Μη ικανοποιητικό = λάθη σε δήλωση, παραμέτρους, κτλ.

Ορθότητα αποτελεσμάτων στην άσκηση 1: Ελέγχουμε αν γίνεται σωστά ο υπολογισμός παραγοντικού και της σειράς. Ιδιαίτερη προσοχή στο αν αρχικοποιούνται σωστά οι μεταβλητές στις οποίες αποθηκεύεται το επιμέρους αποτέλεσμα κατά τη διάρκεια του υπολογισμού (π.χ. το γινόμενο σε 1).

Ένα συχνό λάθος είναι στα όρια της επανάληψης που υπολογίζει το άθροισμα. Το n παίρνει τιμές από 1 (ή 0 ανάλογα με την άσκηση) μέχρι ΚΑΙ το όριο, επομένως γράφουμε for (n=1; n<=limit; n++) και όχι for(n=1; n<limit; n++)

Αλγόριθμος στην άσκηση 2: Και στις δύο ημέρες, προσοχή στα όρια των loops. Ένα string τελειώνει στο '\0', επομένως η αναζήτηση σημείου στίξης πρέπει να γίνει μέχρι εκεί και ΟΧΙ μέχρι SIZE. Ακόμη κι αν δεν επηρεάζει το τελικό αποτέλεσμα, αυτό είναι κάτι που πρέπει να προσέχετε σε προγράμματα που διαχειρίζονται strings. Για παράδειγμα, αν σας ζητούσαμε να μετρήσετε τα σημεία στίξης του string και ελέγχατε και στα σκουπίδια μετά το '\0', μπορεί να βρίσκατε επιπλέον σημεία και να τα μετράγατε χωρίς να είναι σωστό.

[Πέμπτη] Εφόσον εντοπιστεί το σημείο στίξης σε μια θέση i , πρέπει να γράψουμε ένα εσωτερικό loop το οποίο ξεκινά από τη θέση $j=i$ και μέχρι το τέλος του string μεταφέρει το χαρακτήρα της θέσης $j+1$ στη θέση j . Προσοχή να μη βγει εκτός ορίων πίνακα το $j+1$.

Επιπλέον, παρατηρήστε ότι όταν μεταφέρουμε το χαρακτήρα της θέσης $i+1$ μια θέση αριστερά, τότε στη θέση i θα βρίσκεται νέος χαρακτήρας που δεν έχει ακόμη ελεγχθεί, άρα στην επόμενη επανάληψη του εξωτερικού το i δεν πρέπει να προχωρήσει. Γι αυτό, σε αυτή την περίπτωση, κάνουμε ένα $i--$ πριν την επόμενη επανάληψη. Με αυτόν τον τρόπο θα αφαιρεθούν σωστά διαδοχικά σημεία στίξης.

[Παρασκευή] Εφόσον εντοπιστεί το σημείο στίξης σε μια θέση i , πρέπει να γράψουμε ένα εσωτερικό loop το οποίο θα κάνει "χώρο" στη θέση $i+1$. Πρέπει λοιπόν να μεταφέρει όλα τα στοιχεία που ακολουθούν μια θέση δεξιά. Για να γίνει σωστά αυτό πρέπει να ξεκινήσει από το τέλος του string, $j=\text{strlen}(\text{str})$ και να κινηθεί προς τα "αριστερά" μέχρι τη θέση i μεταφέροντας σε κάθε βήμα το χαρακτήρα της θέσης j στη θέση $j+1$. [Η άσκηση υποθέτει ότι τα δεδομένα θα είναι τέτοια ώστε να μη βγούμε εκτός ορίων πίνακα, αλλά κανονικά εδώ θα έπρεπε να προσέξουμε μήπως η θέση $\text{strlen}(\text{str})+1$ είναι εκτός του πίνακα και να κάνουμε κάτι γι αυτό).

Έξοδος: Η έξοδος του προγράμματος πρέπει να είναι ίδια με την εκφώνηση.