ΕΡΩΤΗΜΑ 4

Στην Ασκηση 4 πρέπει να ελέγξουμε αν κάποιες φράσεις είναι παλίνδρομες. Ο τρόπος που υλοποιούμε αυτη τη λειτουργία είναι με τη χρήση ενός stack.

Η main του προγράμματός μου, διαβάζει φράση-φράση από το palindromes.txt. Έχω χρησιμοποιήσει fgets η οποία διαβάζει δεδομένα από το stdin (έχω δώσει ως όρισμα ροής το stdin). Επομένως ο χρήστης πρέπει να κάνει ανακατεύθυνση.

palindromes.c

Είναι το main αρχείο (περιέχει τη main). Μέσα σε μια while, χρησιμοποιώ την fgets και διαβάζω γραμμή γραμμή (κάθε loop και μια διαφορετική φράση). Ουσιαστικά η fgets μου γυρνάει ένα string (δηλ. pointer που δείχνει σε πίνακα χαρ/ων δηλ. string). Το string, αποθηκεύεται στην μεταβλητή str. Σκοπός μου είναι από όλο το string να κρατήσω μόνο τα γράμματα. Γι'αυτό δημιούργησα τη μεταβλητή prestack που είναι επίσης πίνακας χαρ/ων, στην οποία αποθηκεύω ΜΟΝΟ τα γράμματα που υπάρχουν στο str και τα μετατρέπω σε κεφαλαία, σε περίπτωση που είναι πεζά.
Παραδειγμα:

- str = "a B, - Q21 p"
- prestack = "ABQP"

Στη συνέχεια αν ο αριθμός των χαρ/ων του prestack είναι άρτιος καλώ την palEVEN, αλλιώς (αν είναι περιττός) καλώ την palODD (περαιτέρω εξήγηση στο *palindromesMODS.c*). Σε κάθε περίπτωση ως ορίσματα στις συναρτήσεις περνάω το prestack και το πλήθος των χαρ/ων του.

palindromesMODS.c

Το αρχείο αυτό περιλαμβάνει τις συναρτήσεις με τις οποίες επεξεργάζομαι το stack μου (push / pop / init / empty) και ελέγχει αν η φράση είναι παλίνδρομη ή όχι.
Οι συναρτήσεις palEVEN & palODD υλοποιούν την ίδια λειτουργία. Ο λόγος που τις έκανα ξεχωριστές είναι γιατί αν το πλήθος των χαρ/ων είναι περιττό, τότε δεν μας ενδιαφέρει το μεσαίο στοιχείο. Γι'αυτό, χωρίς βλάβη της γενικότητας θα μιλήσω για την palEVEN.
Κάνω τα μίσα στοιχεία Push μέσα στο stack μου (χαρ/ρα - χαρ/ρα). Στη συνέχεια ελέγχω τα υπόλοιπα μισά με τον εξής τροπο:
Κάνω Pop το πάνω πάνω στοιχείο -> Ελέγχω αν είναι ίδιο με τον αντίστοιχο χαρ/ρα του πίνακα (από τη μέση και μετά). ->

- Αν είναι το ίδιο, κάνουμε τον ίδιο έλεγχο με τον επόμενο χαρ/ρα του πίνακα μέχρις ότου να φτάσουμε στο τέλος του πίνακα
- Αν βρεθεί έστω και ένα λάθος (δηλ. ο χαρ/ρας που θα γίνει Pop δεν είναι ίδιος με τον αντίστοιχο χαρ/ρα της αντίστοιχης θέσης) τότε η φράση δεν είναι παλίνδρομη.

Έτσι, σε περίπτωση που δεν υπάρξει κάποιο λάθος, δηλαδή η φράση είναι παλίνδρομη, οι συναρτήσεις γυρνάνε την τιμή "0", διαφορετικά την τιμή "1". Ανάλογα με την τιμή που γυρίσουν εμφανίζεται και το κατάλληλο μήνυμα.

palmods.h / item.h

Δηλώσεις των συναρτήσεων που χρησιμοποιούνται και typedef το char σε Item.

Το makefile μου έχει τις εξής λειτουργίες:

- make
    - gcc -c palindromes.c
    - gcc -c palindromesMODS.c
    - gcc -o palindromes palindromes.o palindromesMODS.o
        - Δημιουργεί τα αντικειμενικά (.o) και στη συνέχεια κάνει τη σύνδεση και μας δίνει το εκτελέσιμο (palindromes)
- make clean
    - rm *.o palindromes
        - Εκτελούμε αυτή την εντολή σε περίπτωση που θέλουμε να σβήσουμε τα αντικειμενικά (.o) και το εκτελέσιμο (palindromes).

Για την εκτέλεση του προγράμματος αρκεί να γράψετε:

- make
- ./palindromes < palindromes.txt

Όπου *palindromes.txt* είναι το αρχείο που πρέπει να ελέγχει το πρόγραμμα μας.

Σημείωσεις:

1. *Έχουν χρησιμοποιηθεί οι συναρτήσεις που βρίσκονται στο αρχείο STACKimplementation.c του Sedgewick.*
2. *Το πρόγραμμα θα μπορούσα να το σπάσω σε ακόμα περισσότερα αρχεία (πχ να είχα σε ένα αρχείο τη main, σε ενα την fQueue και σε ενα τις συναρτήσεις επεξεργασίας του stack απλά το θεώρησα περιττό σε αυτή τη φαση.*