

Άσκηση 3: Χρήση της Στοιβάς για έλεγχο παρενθέσεων σε μαθηματικές εξισώσεις.

Σε αυτή την άσκηση καλείστε να υλοποιήσετε έναν αλγόριθμο, ο οποίος θα δέχεται από την γραμμή εντολών μία εξίσωση (χωρίς κενά – no space characters). Στη συνέχεια, με την υλοποίηση μιας δικής σας Στοιβάς (σε περιοχή της μνήμης που είναι εκτός της καθορισμένης από το SPIM περιοχής) και των δικών σας συναρτήσεων push και pop θα πρέπει να κάνετε χρήση της Στοιβάς και να ελέγχετε εάν τα ζεύγη παρενθέσεων όλων των τύπων { [()] } βρίσκονται σε σωστή αλληλουχία. Για την υλοποίηση της άσκησης που σας ανατίθεται πρέπει να καθορίσετε άλλο χώρο στη μνήμη πέραν της καθορισμένης Στοιβάς του SPIM. Κατά συνέπεια, πρέπει να ορίσετε:

1. Τα όρια της μνήμης που θα κατέχει η δομή,
2. Κάποιο καταχωρητή σαν \$bp και κάποιον σαν \$sp
3. Το μέγιστο μέγεθος της Στοιβάς
4. Επίσης, θα πρέπει να υλοποιηθούν οι βασικές λειτουργίες της δομής (push, pop) καθώς και οι ζητούμενες πράξεις.

Παραδείγματα για σκοπούς επαλήθευσης:

1. Από την γραμμή εντολών δίνετε την εντολή:

```
spim -file filename.s "f=10*{80/[(8+2)-(4/2)]}"
```

Το πρόγραμμα θα πρέπει να σας εκτυπώσει το μήνυμα επιβεβαίωσης ότι η εξίσωση που δώσατε είναι σωστή ως προς τα ζεύγη παρανθέσεων!

2. Εάν αλλάξετε την εξίσωση ώστε να μην υπάρχει σωστό κλείσιμο στις παρενθέσεις κάποιου τύπου δίνοντας την εντολή:

```
spim -file filename.s "f=10*[(8+2)]"
```

τότε το πρόγραμμα θα πρέπει να εκτυπώνει μήνυμα το οποίο θα περιγράφει το είδος του λάθους. Θα πρέπει συνολικά να έχετε τουλάχιστον 4 διαφορετικά μηνύματα λάθους: λάθος κλείσιμο, πλεόνασμα ανοικτών, έλλειψη ανοικτών και μη ύπαρξη παρενθέσεων στην εξίσωση σας.

Απαιτούμενες (Ελάχιστες) Προδιαγραφές Κώδικα:

1. Το πρόγραμμα πρέπει να δέχεται μια εξίσωση (σε μορφή string) από την γραμμή εντολών, η οποία θα αποθηκεύεται στη μνήμη (όπως και στην άσκηση 2).
2. Ο κώδικας σας θα ελέγχει ένα-ένα τα ψηφία που δόθηκαν από τη γραμμή εντολών και **α)** όταν ανιχνεύει οποιαδήποτε αριστερή παρένθεση (δηλαδή άνοιγμα παρένθεσης) να πραγματοποιεί push (εάν η στοιβα βρίσκεται εντός των προκαθορισμένων ορίων), **β)** όταν ανιχνεύει δεξιά παρένθεση να κάνει pop για να ελέγξει εάν κλείνει κανονικά κάποιο ζεύγος παρενθέσεων.
3. Οι συναρτήσεις push και pop πρέπει να διαχειρίζονται λέξεις των 32 μπιτς.
4. Η Στοιβα θα μπορεί να κρατά μέχρι 8 παρενθέσεις ανά πάσα στιγμή. Αν χρειαστεί να αποθηκευτεί 9^η παρένθεση, τότε θα εκτυπώνεται μήνυμα ότι η Στοιβα είναι εκτός ορίων και το πρόγραμμα θα τερματίζει.
5. Στο τέλος του προγράμματος θα πρέπει να εκτυπώνεται ο συνολικός αριθμός των σωστών παρενθέσεων μέχρι το σημείο που εντοπίστηκε το λάθος (νοούμενου ότι υπάρχει κάποιο λάθος).
6. Πρέπει να χρησιμοποιηθούν μόνο οι καταχωρητές \$a0, \$a1 και \$v0 για μεταφορά τιμών προς και από τις συναρτήσεις push και pop.

**** Οποιαδήποτε λύση της άσκησης με αναβάθμιση των προδιαγραφών, θα παίρνει επιπλέον μονάδες!**

Παράδοση και Εξέταση Άσκησης

Η άσκηση θα παραδοθεί σε ηλεκτρονική μορφή (με ηλ. ταχυδρομείο) στον κ. Γιώργο Ζάγγουλο (zaggoulos.george@ucy.ac.cy) με τίτλο email askisi-3 και το αρχείο "your_name.s" συνημμένο. (Το όνομα, ο αριθμός ταυτότητας και η ημέρα εργαστηρίου σας πρέπει να γραφούν τόσο μέσα στο αρχείο κώδικα όσο και στο ηλ. ταχυδρομείο.)

Επίσης, θα πρέπει να παραδοθεί εκτύπωση του κώδικα σας, συνοδευόμενη από μια σύντομη αναφορά (μέχρι 1 σελίδα) και ένα διάγραμμα ροής (flow chart).

Η ημερομηνία και ώρα παράδοσης/αποστολής της άσκησης καθορίζεται ως η μέρα και ώρα έναρξης του εργαστηρίου σας. Για τον λόγο αυτό, θα σημειωθούν οι ημερομηνίες του email server.

Η γραπτή εξέταση θα γίνει την ημέρα παράδοσης της άσκησης. Φοιτητές που δεν θα προσέλθουν στην εξέταση, αυτόματα μηδενίζονται στην άσκηση, ενώ όσοι δεν παραδώσουν την άσκηση, μπορούν να παρακαθήσουν κανονικά στην εξέταση.