



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
www.cslab.ece.ntua.gr

1η ΑΣΚΗΣΗ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ
Ακ. έτος 2013-2014, 5ο Εξάμηνο, Σχολή ΗΜ&ΜΥ

Τελική Ημερομηνία Παράδοσης: 12/01/2014

ΜΕΡΟΣ Α

Δίνεται το παρακάτω πρόγραμμα γραμμένο σε C, καθώς και η αντίστοιχη μετάφραση του σε assembly MIPS. Συμπληρώστε τα κενά. Σας υπενθυμίζουμε ότι ο καταχωρητής \$zero είναι πάντα μηδέν.

```
int exp(int x, int n) {
    if (n == 0) return 1;
    return x * exp (x, n-1);
}

exp:    addi $sp, $sp, ____
        sw   $ra, ____ ($sp)
        sw   $a0, 0($sp)
        bne $a1, ____, cont
        addi ____, $zero, ____
        addi $sp, $sp, ____
        jr   $ra
cont:   addi ____, $a1, -1
        jal  exp
        lw   ____, ____ ($sp)
        mul  ____, $v0, ____
        lw   ____, 4($sp)
        addi $sp, $sp, ____
        jr   $ra
```

ΜΕΡΟΣ Β

Το πρόγραμμα στο μέρος Α υπολογίζει το x^n για ακεραίους $n \geq 0$. Ο ίδιος υπολογισμός μπορεί να πραγματοποιηθεί με 2 ακόμα τρόπους όπως φαίνεται στους παρακάτω ψευδοκώδικες.

Αλγόριθμος I (iterative solution)

```
int exp(int x, int n) {
    for i = 0 .. n
        result *= x;
    return result;
}
```

Αλγόριθμος II (recursive solution version 2)

```
int exp(int x, int n) {
    if (n == 0) return 1;
    else if (n is even)
        return exp(x*x, n/2);
    else
        return x*exp(x*x, (n-1)/2);
}
```

Υλοποιήστε τις αντίστοιχες διαδικασίες σε assembly MIPS, οι οποίες θα επιστρέφουν στον καταχωρητή \$v0 το αποτέλεσμα x^n . Υποθέστε ότι το n είναι θετικός ακέραιος και ότι δε χρειάζεται να ελέγχετε αν ο χρήστης δίνει μη επιτρεπτές τιμές (π.χ. αρνητικές).

ΜΕΡΟΣ Γ

(i) Η *επιθεματική σημειογραφία (postfix notation)* είναι μια μαθηματική σημειογραφία στην οποία οι τελεστές τοποθετούνται μετά από τους τελεστέους τους. Για παράδειγμα, η έκφραση “1 2 +” αντιστοιχεί στην πράξη $1+2=3$. Ομοίως, η έκφραση “1 2 + 3 5 - *” αντιστοιχεί στην πράξη $(1+2)*(3-5)=-6$. Ένας εύκολος τρόπος να τυποποιηθεί ο υπολογισμός μιας έκφρασης σε postfix notation είναι με τη χρήση μιας στοίβας. Κάθε φορά που συναντάται ένας τελεστέος γίνεται push στη στοίβα, ενώ κάθε φορά που συναντάται ένας τελεστής γίνονται pop από τη στοίβα οι τελεστέοι του, εφαρμόζεται σε αυτούς ο τελεστής, και το αποτέλεσμα γίνεται push πίσω στη στοίβα. Ο αριθμός που θα έχει μείνει στη στοίβα μετά τη σάρωση όλων των συμβόλων της έκφρασης και την εκτέλεση των επιμέρους πράξεων θα αντιστοιχεί στην τελική αποτίμηση της αριθμητικής έκφρασης.

Για παράδειγμα:

Σάρωση Έκφρασης	Περιεχόμενα Στοίβας	Λειτουργίες
<u>1</u> 2 + 3 5 - *	1	push(1);
1 <u>2</u> + 3 5 - *	1 2	push(2);
1 2 + <u>3</u> 5 - *	3	x=pop(); y=pop(); z=x+y; push(z);
1 2 + 3 <u>5</u> - *	3 3	push(3);
1 2 + 3 5 <u>-</u> *	3 3 5	push(5);
1 2 + 3 5 - <u>*</u>	3 -2	x=pop(); y=pop(); z=x-y; push(z);
1 2 + 3 5 - * <u>-</u>	-6	x=pop(); y=pop(); z=x*y; push(z);

Στα πλαίσια αυτού του ερωτήματος καλείστε να υλοποιήσετε ένα πρόγραμμα MIPS το οποίο υπολογίζει την αποτίμηση μιας έκφραση σε postfix notation. Χάριν απλότητας έχουμε υποθέσει τα εξής:

- οι τελεστέοι είναι ακέραιοι, μονοψήφιοι και θετικοί αριθμοί
- οι τελεστές είναι οι «+», «-», «*», «/»
- η έκφραση είναι συντακτικά ορθή και ολοκληρωμένη, δηλαδή δεν έχει σφάλματα όπως π.χ. έλλειψη τελεστών (π.χ. 1 2 + 4), έλλειψη τελεστέων (π.χ. 1 2 + -), λανθασμένη σειρά τελεστών-τελεστών (π.χ. + 1 2), κ.ο.κ..

Υλοποιήστε ένα πρόγραμμα σε MIPS assembly το οποίο θα αποτιμά εκφράσεις σε postfix notation με βάση τις παραπάνω παραδοχές. Η αριθμητική έκφραση εισόδου δίνεται στο πρόγραμμα σαν μια ακολουθία από σύμβολα (bytes) σε ASCII κώδικα τα οποία έχουν αποθηκευτεί σε συνεχόμενες θέσεις μνήμης. Η αρχή της ακολουθίας βρίσκεται στη θέση μνήμης 0x1000, ενώ το τέλος της ακολουθίας σηματοδοτείται από τον χαρακτήρα «\$». Το αποτέλεσμα θα πρέπει να αποθηκεύεται στο τέλος στη θέση μνήμης 0x2000.

Για την υλοποίηση της άσκησης μπορείτε να χρησιμοποιήσετε τον **MSIM**, ένα emulator του MIPS ο οποίος αναπτύχθηκε από συμφοιτητές σας και διατίθεται από το εργαστήριο Υπολογιστικών Συστημάτων (CSLab). Στον emulator αυτό, μπορείτε να γράφετε MIPS assembly και να την εκτελείτε παρακολουθώντας τα περιεχόμενα των καταχωρητών και της μνήμης καθιστώντας έτσι ευκολότερη την παραγωγή και τον έλεγχο του απαιτούμενου κώδικα. Τον MSIM μπορείτε να τον κατεβάσετε από τη σελίδα των ασκήσεων του site του μαθήματος.

Παραδοτέο της άσκησης θα είναι ένα ηλεκτρονικό κείμενο (pdf, docx ή odt) που θα περιέχει τους κώδικες assembly και των 3 μερών της άσκησης. Ο κώδικας θα πρέπει να περιέχει αναλυτικά σχόλια για την κατανόηση της λύσης σας από τους διδάσκοντες. Στο ηλεκτρονικό κείμενο να αναφέρετε στην αρχή τα στοιχεία σας (Όνομα, Επώνυμο, ΑΜ).

Η άσκηση θα παραδοθεί ηλεκτρονικά στην ιστοσελίδα:

<http://www.cslab.ece.ntua.gr/courses/comparch/submit-tmimal>

Δουλέψτε ατομικά. Έχει ιδιαίτερη αξία για την κατανόηση του μαθήματος να κάνετε μόνοι σας την εργασία. Μην προσπαθήσετε να την αντιγράψετε από άλλους συμφοιτητές σας.