



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

1η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

Ακ. έτος 2016-2017, 5ο Εξάμηνο, Σχολή ΗΜΜΥ

(τμήμα Μ – Ω)

Ημερομηνία Παράδοσης: 6/11/2016

Απορίες στο: ca2016-2017-tmima2@cslab.ece.ntua.gr

ΜΕΡΟΣ Α

Δίνεται ο ακόλουθος κώδικας σε C, καθώς και η αντίστοιχη μετάφρασή του σε MIPS assembly. Δίνεται επίσης ότι η διεύθυνση του πρώτου στοιχείου του πίνακα **A** βρίσκεται αποθηκευμένη στον καταχωρητή **\$s1**, η διεύθυνση του πρώτου στοιχείου του πίνακα **B** στον καταχωρητή **\$s2**, η διεύθυνση του πρώτου στοιχείου του πίνακα **C** στον καταχωρητή **\$s3**, ενώ ο καταχωρητής **\$s0** περιέχει την τιμή του **N**. Θεωρήστε πως ο πίνακας **A** περιέχει στοιχεία στο διάστημα **[0, N-1]**. Υπενθυμίζεται τέλος πως ο τύπος **int** της C έχει μέγεθος **4 bytes (1 word)** για την αρχιτεκτονική MIPS). Ζητείται να συμπληρωθούν καταλλήλως τα κενά.

<pre>int A[N], B[N], C[N], i; for (i = 0; i < N; i++) B[i] = A[i] + C[A[i]];</pre>	<pre>add \$t1, ____, ____ # i = 0 LOOP: slt \$t0, \$t1, ____ beq ____, \$zero, END # if (i >= N) goto END sll \$t2, \$t1, ____ # i * 4 add \$t6, \$t2, ____ # calculate Addr(A[i]) ____ \$t3, 0(____) # read A[i] ____ \$t4, ____, 2 # A[i] * 4 add \$t6, \$t4, ____ # calculate Addr(C[A[i]]) lw \$t4, ____ (____) # read C[A[i]] add \$t5, ____, ____ # A[i] + C[A[i]] add \$t6, ____, \$s2 # calculate Addr(B[i]) ____ \$t5, 0(____) # save B[i] addi ____, \$t1, 1 # i++ ____ ____ END:</pre>
--	--

ΜΕΡΟΣ Β

Δίνεται η συνάρτηση σε C που υλοποιεί τον αλγόριθμο του πολλαπλασιασμού αλά ρωσικά (Russian peasant multiplication). Να υλοποιηθεί σε MIPS assembly.

```
unsigned int russian_peasant_mult(unsigned int a, unsigned int b) {  
  
    // initialize result  
    int res = 0;  
  
    // while second number doesn't become 1  
    while (b > 0) {  
  
        // if second number becomes odd, add the first number to result  
        if (b & 1) {  
            res = res + a;  
        }  
  
        // double the first number and halve the second number  
        a = a << 1;  
        b = b >> 1;  
  
    }  
  
    return res;  
  
}
```

ΜΕΡΟΣ Γ

Δοθέντων δύο ακεραίων, n και k , όπου $n \leq k$, επιθυμούμε να βρούμε το **άθροισμα** των ακεραίων από n έως k . Για παράδειγμα, αν $n = 1$ και $k = 5$, το ζητούμενο άθροισμα ισούται με $1 + 2 + 3 + 4 + 5 = 15$. Για το σκοπό αυτό σας δίνονται οι παρακάτω δύο συναρτήσεις, οι οποίες επιτελούν τον ίδιο υπολογισμό. Ποια είναι η διαφορά τους όσον αφορά την υλοποίηση; Δώστε τις αντίστοιχες υλοποιήσεις σε MIPS assembly.

```
int sum1(int n, int k) {  
    int i, res = 0;  
    for (i = n; i <= k; i++) {  
        res += i;  
    }  
    return res;  
}  
  
int sum2(int n, int k) {  
    if (n > k) {  
        return 0;  
    }  
    return n + sum2(n + 1, k);  
}
```

* * *

Σημείωση: Για τη διευκόλυνσή σας, σας προτρέπουμε να χρησιμοποιήσετε το προσομοιωτή Qtspim (<http://spimsimulator.sourceforge.net/>).

Παραδοτέο της άσκησης θα είναι ηλεκτρονικό κείμενο (pdf, doc/docx ή odt) που θα περιέχει τις απαντήσεις των τριών μερών. Δε θα γίνουν δεκτά παραδοτέα σε χειρόγραφή μορφή. Το έγγραφο πρέπει να φέρει τα στοιχεία σας (όνομα, επώνυμο και αριθμό μητρώου). Οι κώδικες που θα παραδοθούν οφείλουν να είναι σε ευανάγνωστη μορφή και να φέρουν επαρκή σχόλια.