



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

1η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ  
Ακ. έτος 2017-2018, 5ο εξάμηνο, ΣΗΜΜΥ

ΤΜΗΜΑ 2ο (Μ - Ω)  
Ημερομηνία παράδοσης: 26/11/2017  
Απορίες στο: ca2017-2018-tmima2@cslab.ece.ntua.gr

## ΜΕΡΟΣ Ι

Δίνεται η ακόλουθη συνάρτηση γραμμένη σε C, καθώς και η μετάφρασή της σε assembly του MIPS. Να συμπληρωθούν τα κενά. Υπενθυμίζεται πως ο καταχωρητής \$zero περιέχει την τιμή 0, η οποία και δεν μπορεί να αλλάξει. Όσον αφορά τα ορίσματα και την επιστροφή αποτελεσμάτων από τη ρουτίνα, παρακαλείστε να ακολουθήσετε τις γνωστές από το μάθημα συμβάσεις.

```
int fib(int n) {  
    int i = 1, j = 0, k, t;  
    for (k = 1; k <= n; k++) {  
        t = i + j;  
        i = j;  
        j = t;  
    }  
    return j;  
}  
  
fib:  addi $t1, ___, 1  
      add $t2, $zero, ___  
      addi $t3, $zero, ___  
      addi $t5, ___, 1  
loop: slt ___, $t3, $t5  
      beq $t0, ___, exit  
      add $t4, $t2, ___  
      add $t1, $t2, ___  
      add ___, $t4, ___  
      addi $t3, ___, ___  
      j   ___  
exit: add ___, ___, $t2  
      jr  ___
```

## ΜΕΡΟΣ ΙΙ

Να γραφεί ρουτίνα σε assembly του MIPS που να πραγματοποιεί **in-place** αντιστροφή της θέσης των στοιχείων πίνακα ακεραίων αριθμών **32-bit**. Για παράδειγμα, όταν δοθεί ως όρισμα ο πίνακας [13, 100, -2], ο κώδικάς σας θα πρέπει να τον μετασχηματίζει σε [-2, 100, 13]. Τα ορίσματά της θα είναι η διεύθυνση του πρώτου στοιχείου καθώς και το μέγεθος του πίνακα (σε αριθμό στοιχείων — **όχι** σε bytes).

## ΜΕΡΟΣ III

Δίνεται η ακόλουθη συνάρτηση γραμμένη σε C, η οποία υλοποιεί τον αλγόριθμο της δυαδικής αναζήτησης σε ταξινομημένο πίνακα ακεραίων **32-bit** (binary search algorithm<sup>1</sup>). Να αποδοθεί η ισοδύναμή της σε assembly του MIPS.

```
int binary_search(int *arr, int l, int r, int x) {
    if (r >= l) {
        int mid = l + (r - l) / 2;

        // if the element is present
        // at the middle itself
        if (arr[mid] == x) return mid;

        // if element is smaller than mid,
        // then it can only be present in left subarray
        if (arr[mid] > x) return binary_search(arr, l, mid-1, x);

        // else the element can only be present
        // in right subarray
        return binary_search(arr, mid+1, r, x);
    }
    // we only reach here when element
    // is not present in array
    return -1;
}
```

\* \* \*

**Σημείωση:** Για διευκόλυνση, σας προτρέπουμε να χρησιμοποιήσετε το προσομοιωτή Qtspim (<http://spimsimulator.sourceforge.net>). Εδώ μπορείτε να βρείτε ένα λεπτομερές εγχειρίδιο χρήσης του.

Παραδοτέο της άσκησης θα είναι ένα **ηλεκτρονικό κείμενο** (κατά προτίμηση **pdf**, για λόγους συμβατότητας) που θα περιέχει τις απαντήσεις των τριών μερών. Το έγγραφο πρέπει να φέρει στην αρχή του τα στοιχεία σας (όνομα, επώνυμο και αριθμό μητρώου). Οι κώδικες που θα παραδοθούν οφείλουν να είναι σε ευανάγνωστη μορφή και να φέρουν επαρκή σχόλια.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Binary\\_search\\_algorithm](https://en.wikipedia.org/wiki/Binary_search_algorithm)