



## 1η ΑΣΚΗΣΗ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

Ακ. έτος 2022-2023, 5ο Εξάμηνο, Σχολή ΗΜ&ΜΥ

Τελική Ημερομηνία Παράδοσης: 27/11/2022

### ΜΕΡΟΣ Α

Δίνεται το παρακάτω πρόγραμμα σε C καθώς και μια μετάφραση του σε assembly MIPS. Θεωρήστε ότι ο καταχωρητής \$s0 περιέχει τη διεύθυνση του πρώτου στοιχείου του πίνακα array. Συμπληρώστε τα κενά. Σας υπενθυμίζουμε ότι ο καταχωρητής \$0 (ή \$zero) είναι πάντα μηδέν.

<pre>int array[20]; int *p, N;  p = &amp;array[3];  while (*p != 0) {     if (*p &lt; 50) *p = *p % N;     else *p = *p / N;     p++; }</pre>	<pre>LOOP:     addi    _____, \$s0, _____     lw      _____, 0(\$s1)     _____ \$t0, _____, _____     div     _____, \$s2     _____ \$t0, 50     _____ \$t1, _____, _____     _____ \$t0     jmp     _____ ELSE: NEXT:     sw      _____, _____     addi    _____, _____, _____     jmp     _____ END:</pre>
---	--

### ΜΕΡΟΣ Β

Οι παρακάτω ρουτίνες σε C υλοποιούν τους αλγορίθμους ταξινόμησης **bubble sort**, **cocktail sort** και **insertion sort** ενός πίνακα ακεραίων με N στοιχεία. Υλοποιήστε τις ρουτίνες σε assembly του MIPS.

```
void bubbleSort(int *A, int N) {
    int i, j;
    bool swapped;

    for (i = 0; i < N-1; i++) {
        swapped = false;
        for (j = 0; j < N-i-1; j++)
            if (A[j] > A[j+1]) {
                swap(&A[j], &A[j+1]);
                swapped = true;
            }
        if (!swapped) break;
    }
}
```

```

void cocktailSort(int *A, int N) {
    bool swapped = true;
    int start = 0;
    int end = N-1;

    while (swapped) {
        swapped = false;
        for(int i = start; i < end; i++) {
            if (A[i] > A[i+1]) {
                swap(&A[i], &A[i+1]);
                swapped = true;
            }
        }
        if (!swapped) break;

        swapped = false;
        end--;
        for(int i = end - 1; i >= start; i--) {
            if(A[i] > A[i+1]) {
                swap(&A[i], &A[i+1]);
                swapped = true;
            }
        }
        start++;
    }
}

void insertionSort(int *A, int N) {
    int key, j;

    for(int i = 1; i < N; i++) {
        key = A[i];
        j = i - 1;

        while (j >= 0 && A[j] > key) {
            A[j+1] = A[j];
            j--;
        }
        A[j+1] = key;
    }
}

void swap (int *x, int * y) {
    int temp = *x;

    *x = *y;
    *y = temp;
}

```

### ΜΕΡΟΣ Γ

Υλοποιήστε ξανά τον αλγόριθμο ταξινόμησης insertion sort χρησιμοποιώντας αυτή τη φορά αναδρομή.

Για την υλοποίηση της άσκησης μπορείτε να χρησιμοποιήσετε τον **MILE**, ένα MIPS emulator που αναπτύχθηκε από συμφοιτητές σας και διατίθεται από το Εργαστήριο Υπολογιστικών Συστημάτων (<http://www.cslab.ece.ntua.gr/courses/comparch/assign.go>). 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.

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 στο moodle του μαθήματος:

<https://helios.ntua.gr/course/view.php?id=1038>

*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.*