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

1η ΑΣΚΗΣΗ ΣΤΗΝ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΥΠΟΛΟΓΙΣΤΩΝ

Ακ. έτος 2016-2017, 5ο Εξάμηνο, Σχολή ΗΜ&ΜΥ

Τελική Ημερομηνία Παράδοσης: **13/11/2016**

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 i, j, tmp;
int *arr, n;

for (i=0; i < n; i++){
   for (j=0; j < n-i-1; j++){
      if (arr[j] > arr[j+1]) {
         tmp      = arr[j];
         arr[j]   = arr[j+1];
         arr[j+1] = tmp;
      }
   }
}
```

```
                add  $t0,$zero,_____
I_LOOP:         beq  ___,$s3, END
                add  $t1,_____,$zero
J_LOOP:         sub  ___, $s3, $t0
                addi $t2, $t2, _____
                beq  $t2, $t1, _____
                sll  ___, $t1, _____
                add  $t2, ___, $s1
                lw   $t3, __($t2)
                lw   ___, __($t2)
                slt  $t5, ___, ____
                beq  $t5, ___,_____
                sw   $t4, __($t2)
                sw   ___, __($t2)
NEXT_J:         addi $t1, ___, ____
                jmp  _____
NEXT_I:         addi $t0, ___, ____
                jmp  _____
END:
```

ΜΕΡΟΣ Β

Οι παρακάτω C ρουτίνες υλοποιούν τον αλγόριθμο **xorshift**[1], ο οποίος παράγει ψευδοτυχαίους αριθμούς. Η συνάρτηση **xorshift_init()** καλείται μία φορά για την αρχικοποίηση των global μεταβλητών x, y, z, w και στη συνέχεια κάθε κλήση της **xorshift()** επιστρέφει έναν τυχαίο 32-bit αριθμό. Υλοποιήστε τις ρουτίνες σε assembly του MIPS, υποθέτοντας πως οι μεταβλητές x, y, z, w αποθηκεύονται σε συνεχόμενες θέσεις της μνήμης ξεκινώντας από την 0x10000000.

```
uint32_t x, y, z, w;

void xorshift_init(int _x, int _y, int _z, int _w) {
      x = _x;
      y = _y;
      z = _z;
      w = _w;
}
```

[1] https://en.wikipedia.org/wiki/Xorshift

```
uint32_t xorshift()
{
      uint32_t t = x;
      t ^= t << 11;
      t ^= t >> 8;
      x = y; y = z; z = w;
      w ^= w >> 19;
      w ^= t;
      return w;
}
```

ΜΕΡΟΣ Γ

Οι παρακάτω C ρουτίνες υλοποιούν τον αλγόριθμο ταξινόμησης **quicksort**[2] για ακέραιους αριθμούς 32-bit. Η quicksort χρησιμοποιεί την xorshift() για την παραγωγή ενός τυχαίου αριθμού μεταξύ left και right τον οποίο χρησιμοποιεί σαν pivot. Υλοποιήστε τις ρουτίνες σε assembly του MIPS.

```
void swap(int *m1, int *m2) {
      int tmp = *m1;

      *m1 = *m2;
      *m2 = tmp;
}
int partition(int *A, int left, int right) {
      int pivot, i, j;

      pivot = A[left];
      i = left - 1;
      j = right + 1;

      while (1) {
            while (A[++i] < pivot);
            while (A[--j] > pivot);

            if (i < j)
                  swap(&A[i], &A[j]);
            else
                  return j;
      }
}
void quicksort(int *A, int left, int right) {
      if (left >= right) return;

      int pivot = xorshift128() % (right - left) + left;
      swap(&A[pivot], &A[left]);

      int q = partition(A, left, right);
      quicksort(A, left, q);
      quicksort(A, q+1, right);
}
```

[2] https://en.wikipedia.org/wiki/Quicksort

Για την υλοποίηση της άσκησης μπορείτε να χρησιμοποιήσετε τον ***MSIM***, ένα MIPS emulator που 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-tmima1

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.