



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Θεμελιώδη Θέματα Επιστήμης Υπολογιστών, 2018-19

Διδάσκων: Γ. Γκούμας

4η σειρά ασκήσεων (προαιρετική)

(Προγραμματισμός στη γλώσσα assembly του LC3)

Οι “κακοί αριθμοί” (evil numbers) είναι θετικοί αριθμοί, των οποίων η δυαδική αναπαράσταση έχει άρτιο αριθμό από 1. Για παράδειγμα, ο αριθμός 10 είναι κακός αριθμός, αφού η δυαδική του αναπαράσταση είναι 1010 με 2 ψηφία ίσα με 1, ενώ ο αριθμός 11 δεν είναι, αφού η δυαδική του αναπαράσταση είναι 1011 με 3 ψηφία ίσα με 1. Ακολουθώντας, σας δίνεται μία συνάρτηση σε γλώσσα C που δέχεται ως είσοδο έναν αριθμό x και επιστρέφει 1 αν ο αριθμός είναι “κακός” και 0 αλλιώς.

```
1 int is_evil (int x) {
2   //η μεταβλητή sum κρατά το άθροισμα των δυαδικών ψηφίων
3   int sum = 0;
4   int r;
5   //μετατροπή του x σε δυαδικό
6   while (x != 0) {
7     r = x % 2;      //υπόλοιπο διαίρεσης με 2
8     x = x / 2;
9     sum += r;       //πρόσθεση δυαδικού ψηφίου στο sum
10  }
11
12  if (sum % 2 == 0) //αν sum άρτιος
13    return 1;      //ο αριθμός είναι evil – επέστρεψε 1
14  else
15    return 0;      //αλλιώς επέστρεψε 0
16 }
```

Σας δίνεται στη συνέχεια ένα ημιτελές πρόγραμμα στη γλώσσα assembly του LC3. Στη διεύθυνση μνήμης DATA (x5000) έχει αποθηκευτεί ένας πίνακας από αριθμούς μεγέθους LENGTH. Το πρόγραμμά σας θα πρέπει να διαβάζει καθέναν από αυτούς τους αριθμούς, να υπολογίζει αν είναι “κακός” ή όχι και να αποθηκεύει το αποτέλεσμα σε έναν πίνακα ίδιου μήκους, στη διεύθυνση RESULTS (x6000).

Για την υλοποίηση του προγράμματός σας, καλείστε να υλοποιήσετε δύο υπορουτίνες. Η υπορουτίνα IS_EVIL δέχεται ως είσοδο έναν αριθμό x και επιστρέφει 1, αν αυτός είναι “κακός” και 0 αλλιώς. Αυτή ισοδυναμεί με τη συνάρτηση is_evil που σας δόθηκε παραπάνω σε C. Η υπορουτίνα DIV_MOD_2 δέχεται ως είσοδο έναν αριθμό x και υπολογίζει το πηλίκο της διαίρεσης του αριθμού με το 2 και το υπόλοιπο της διαίρεσης του αριθμού με το 2. Αυτή η υπορουτίνα θα υλοποιεί τις γραμμές 7 και 8 του κώδικα που σας δίνεται παραπάνω και θα καλείται από την υπορουτίνα IS_EVIL για την εύρεση της δυαδικής αναπαράστασης του κάθε αριθμού. Θα καλείται επίσης από την υπορουτίνα IS_EVIL για να υπολογίζει αν το άθροισμα των δυαδικών ψηφίων είναι άρτιο ή περιττό.

Συμπληρώστε το πρόγραμμα (τόσο το κυρίως πρόγραμμα όσο και τις δύο υπορουτίνες). Στη διεύθυνση μνήμης DATA (x5000) μπορείτε να αποθηκεύσετε δεδομένα εισόδου μέσω αρχείου. Ένα παράδειγμα για LENGTH = 10 σας δίνεται στο αρχείο data.asm. Το πρόγραμμά σας θα πρέπει να διαβάζει καθέναν από αυτούς τους αριθμούς, να υπολογίζει αν είναι “κακός” ή όχι και να αποθηκεύει το αποτέλεσμα

σε έναν πίνακα ίδιου μήκους, στη διεύθυνση RESULTS (x6000). Το πρόγραμμά σας θα πρέπει να περιλαμβάνει σχόλια.

```
.ORIG x3000                                ; Program that calculates if
                                           ; the numbers stored in array
                                           ; starting at address DATA
                                           ; are evil or not.
                                           ; Results for each number
                                           ; are stored in an array
                                           ; starting at address RESULTS.
                                           ; Calls the IS_EVIL subroutine.

      LDI R4, DATA_ADDR                  ; R4 contains address of DATA
      LDI R5, RESULTS_ADDR                ; R5 contains address of RESULTS
      LD R6, LENGTH                       ; R6 = LENGTH

REPEAT LDR R0, R4, x0000                  ; R0 = M[R4]

...

END          HALT
```

```
IS_EVIL ; Subroutine to compute if a number
        ; is evil.
        ; INPUT number in R0.
        ; OUTPUT result in R1
        ; (if R0 evil, then R1=1, else R1=0)
        ; Calls DIV_MOD_2 subroutine
```

...

```
DIV_MOD_2 ; Subroutine to compute
          ; quotient and remainder
          ; of division by 2.
          ; INPUT dividend in R0.
          ; OUTPUT remainder in R2 (R2 = R0 % 2)
          ; OUTPUT quotient in R0 (R0 = R0 / 2)
```

...

```
LENGTH .FILL 100  
DATA_ADDR .FILL DATA  
RESULTS_ADDR .FILL RESULTS
```

```
.BLKW 1000
```

```
DATA .FILL x5000  
RESULTS .FILL x6000
```

```
.END
```