10

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 5 – ΟΜΑΔΑ Α5

Μηλτιάδης Μαντές AM 1084661 EMAIL up1084661@upnet.gr

Ελπίδα Κόκκαλη AM 1084648 EMAIL up1084648@upnet.gr

i. Υλοποίηση Insertion sort/in-place

```
.arm
.text
.global main

main:
STMDB R13!, {R1,R12}
LDR R0, =Table                  @Θέτουμε τον R1 να δείχνει στις διευθύνσεις του πίνακα Table
MOV R1, #6                      @Μεταφέρουμε στον R1 τη τιμή 6 και τον χρησιμοποιούμε για να διαβάζουμε τα bytes στις διευθύνσεις των πινάκων
BL Subrtn                       @Καλούμε την υπορουτίνα 1 φορά
LDMIA R13!, {R1-R12}            @Επανακτούμε το περιεχόμενο των καταχωρητών που είχαμε σώσει


Subrtn:
STMDB R13!, {R2-R7}
MOV R3, #0                      @Μεταφέρουμε στον R3 τη τιμή 0 και τον χρησιμοποιούμε για να διαβάζουμε τα bytes στις διευθύνσεις του πίνακα


Startpoint1:
ADD R5, R3, #1                  @Αυξάνουμε τη τιμή του R3 κατά 1 και αποθηκεύουμε το αποτέλεσμα στον R5 (1)
LDRB R2, [R0, R3]               @Μεταφέρουμε στον R2 το byte της πρώτης διεύθυνσης μνήμης όπου δείχνει ο R0 (δηλαδή τον πρώτο αριθμό κ.ο.κ στη συνέχεια)
MOV R4, R3                      @Μεταφέρουμε στον R4 τη τιμή του καταχωρητή R3


Startpoint2:
LDRB R6, [R0, R5]               @Μεταφέρουμε στον R6 το byte της δεύτερης διεύθυνσης μνήμης όπου δείχνει ο R0 (δηλαδή τον δεύτερο αριθμό κ.ο.κ στη συνέχεια)
```

```
CMP R6, R2                  @Συγκρίνουμε τα περιεχόμενα των καταχωρητών R2, R6 (δηλαδή τον πρώτο με τον δεύτερο αριθμό του πίνακα κ.ο.κ στη συνέχεια)
MOVLO R2, R6                @Αν το περιεχόμενο του R6 είναι μικρότερο από αυτό του R2 (δηλαδή ο δεύτερος αριθμός είναι μικρότερος από τον πρώτο) τότε αποθηκεύουμε τον μικρότερο στον καταχωρητή R2
ADD R5, R5, #1              @Αυξάνουμε το περιεχόμενο του R5 κατά 1 για να προσπελάσουμε το στοιχείο από την επόμενη θέση μνήμης από όπου δείχνει ο καταχωρητής R0
CMP R5, #6                  @Συγκρίνουμε το περιεχόμενο του R5 με το 6 για να πραγματοποιηθούν 6 επαναλήψεις
BLO Startpoint2             @Εκτελούμε την υπορουτίνα Startpoint2 μέχρι το περιεχόμενο του R5 να γίνει 6
LDRB R7, [R0, R3]           @Μεταφέρουμε στον R7 το byte της πρώτης διεύθυνσης μνήμης όπου δείχνει ο R0 (δηλαδή τον πρώτο αριθμό κ.ο.κ στη συνέχεια)
STRB R2, [R0, R3]           @Φορτώνουμε επίσης στον R2 το byte της πρώτης διεύθυνσης μνήμης όπου δείχνει ο R0 (δηλαδή τον πρώτο αριθμό κ.ο.κ στη συνέχεια)
STRB R7, [R0, R4]           @Φορτώνουμε επίσης στον R7 το byte της πρώτης διεύθυνσης μνήμης όπου δείχνει ο R0 (δηλαδή τον πρώτο αριθμό κ.ο.κ στη συνέχεια)
ADD R3, R3, #1              @Αυξάνουμε το περιεχόμενο του R5 κατά 1 για να προσπελάσουμε το στοιχείο από την επόμενη θέση μνήμης από όπου δείχνει ο καταχωρητής R0
CMP R3, #5                  @Συγκρίνουμε το περιεχόμενο του R3 με το 5 για να πραγματοποιηθούν άλλες 5 επαναλήψεις
BLO Startpoint1             @Εκτελούμε την υπορουτίνα Startpoint1 μέχρι το περιεχόμενο του R3 να γίνει 5
LDMIA R13!, {R2-R7}         @Επανακτούμε το περιεχόμενο των καταχωρητών που είχαμε σώσει
MOV PC, LR                  @Επιστρέφουμε από την υπορουτίνα στο σημείο όπου κλήθηκε
```

```
Startpoint3:
LDRB R2,[R0,R4]                 @Μεταφέρουμε στον R2 το byte της πρώτης διεύθυνσης μνήμης όπου δείχνει ο R0 (δηλαδή τον πρώτο αριθμό κ.ο.κ στη συνέχεια)
ADD R4,R4 , #1                  @Αυξάνουμε το περιεχόμενο του R5 κατά 1 για να προσπελάσουμε το στοιχείο από την επόμενη θέση μνήμης από όπου δείχνει ο καταχωρητής R0
LDRB R3,[R0,R4]                 @Μεταφέρουμε στον R3 το byte της δεύτερης διεύθυνσης μνήμης όπου δείχνει ο R0 (δηλαδή τον δεύτερο αριθμό κ.ο.κ στη συνέχεια)
CMP R3,R2                       @Συγκρίνουμε τα περιεχόμενα των καταχωρητών R3, R2 (δηλαδή τον πρώτο με τον δεύτερο αριθμό του πίνακα κ.ο.κ στη συνέχεια)
MOVLO R5,#false                 @Αν το περιεχόμενο του R2 είναι μικρότερο από αυτό του R3 (δηλαδή ο πρώτος αριθμός είναι μικρότερος από τον δεύτερο) τότε αποθηκεύουμε στον καταχωρητή R5 τη τιμή false
ADD R4,R4,#1                    @Αυξάνουμε το περιεχόμενο του R5 κατά 1 για να προσπελάσουμε το στοιχείο από την επόμενη θέση μνήμης από όπου δείχνει ο καταχωρητής R0
CMP R4,#5                       @Συγκρίνουμε το περιεχόμενο του R3 με το 5 για να πραγματοποιηθούν άλλες 5 επαναλήψεις
BLO Startpoint3                 @Εκτελούμε την υπορουτίνα Startpoint3 μέχρι το περιεχόμενο του R4 να γίνει 5
STRB R5,[R6]                    @Φορτώνουμε στον R6 την τιμή του καταχωρητή R5
LDMIA R13!, {R2-R5}
MOV PC,LR

.data
Table:
.byte 0x45, 0x82, 0x34, 0xDA, 0x10, 0x28
```

ii. Εκτέλεση αλγορίθμου και επιβεβαίωση ορθότητας αποτελεσμάτων

```
.arm
.text
.global main
.equ true, 1
.equ false, 0

main:
STMDB R13!, {R1-R12}
```

```
LDR R0, = Table
MOV R1, #20
LDR R6, =flag
BL Subrtn
BL Startpoint3
LDMIA R13! , {R1-R12}

Subrtn:
STMDB R13!, {R2-R7}
MOV R3, #0

Startpoint1:
ADD R5, R3, #1
LDRB R2,[R0,R3]
MOV R4,R3

Startpoint2:
LDRB R6, [R0,R5]
CMP R6,R2
MOVLO R4,R5
MOVLO R2,R6
ADD R5,R5, #1
CMP R5,#20
BLO Startpoint2
LDRB R7, [R0,R3]
STRB R2,[R0,R3]
STRB R7, [R0,R4]
ADD R3,R3, #1
CMP R3,#19
BLO Startpoint1
LDMIA R13!, {R2-R7}

Startpoint3:
STMDB R13!, {R2-R5}
```

```
MOV R3,#0
MOV R5, #true

Startpoint4:
LDRB R2,[R0,R4]
ADD R4,R4 , #1
LDRB R3,[R0,R4]
CMP R3,R2
MOVLO R5,#false
ADD R4,R4,#1
CMP R4,#19
BLO Startpoint4
STRB R5,[R6]
LDMIA R13!, {R2-R5}
MOV PC,LR
.data
Table:
.byte 0x31,0x52,0x34,0xFA,0x1F,0xC8,0xB4, 0x98, 0x20, 0x21,
0x02, 0xAA, 0xA1, 0xBA, 0x60, 0x30, 0xCC,0x77, 0xF7, 0xF8,
0xF9

flag:
.byte true
```