

i. Υλοποίηση Insertion sort/in-place

```

0  .arm
1  .text
2  .global main
3
4  main:
5  STMDB R13!, {R0 - R12, R14}
6
7  LDR R0, =Table
8  MOV R1, #6      @table_size
9
10 BL Insertion
11
12 LDMIA R13!, {R0- R12, R14}
13
14
15 Insertion:
16 STMDB R13!, {R0-R12}
17
18 LDRB R2, [R0, #0]  @μικρότερο στοιχείο
19 MOV R3, #0         @θέση πρώτου
20 MOV R4, #0         @θέση μικρότερου
21 MOV R5, #1         @θέση επόμενου
22
23 MainLoop:
24
25 ADD R5, R3, #1
26 LDRB R2, [R0, R3]
27 MOV R4, R3
28
29 Loop:
30
31 LDRB R6, [R0, R5]
32
33 CMP R2, R6
34 MOVHI R4, R5
35 MOVHI R2, R6
36
37 ADD R5, R5, #1
38 CMP R5, R1
39 BLT Loop
40
41 LDRB R7, [R0, R3]
42 STRB R2, [R0, R3]
43 STRB R7, [R0, R4]
44 ADD R3, R3, #1
45
46 SUB R8, R1, #1
47
48 CMP R3, R8
49 BLT MainLoop
50
51 LDMIA R13!, {R0-R12}
52 MOV PC, LR
53
54
55 .data
56
57 Table:
58 .byte 0x45, 0x82, 0x34, 0xDA, 0x10,
59 0x28
60
61
62
63
64
65
66
67
68
69

```

Στον R0 αποθηκεύεται η διεύθυνση μνήμης του πίνακα. Στον R1 αποθηκεύεται ο αριθμός των στοιχείων του πίνακα.

Στην υπορουτίνα Insertion χρησιμοποιούμε τον καταχωρητή R2 για την αποθήκευση του μικρότερου στοιχείου που βρίσκουμε μέσω των συγκρίσεων, τον R3 για την θέση του πρώτου αριθμού που συγκρίνουμε, τον R4 για την θέση το μικρότερου και τον R5 για την θέση του επόμενου που θα συγκρίνουμε.

Στην κύρια επανάληψη της υπορουτίνας θα βρίσκουμε το μικρότερο στοιχείο απ' όλο το πίνακα κάθε φορά. Αρχικά αυξάνουμε τον R5 κατά τη θέση του πρώτου + 1, για την επόμενη σύγκριση και αποθηκεύουμε στον R2 το περιεχόμενο του πίνακα που δείχνει ο pointer - R3. Μεταφέρουμε το περιεχόμενο του R3 (τη θέση του πρώτου) στον R4 (στη θέση του μικρότερου, αφού θα αντικατασταθεί με τον μικρότερο).

Στην εσωτερική επανάληψη συγκρίνουμε το πρώτο στοιχείο με όλα τα υπόλοιπα. Αποθηκεύουμε στον καταχωρητή R6 το περιεχόμενο του επόμενου στοιχείου και το συγκρίνουμε με τα άλλα για να βρούμε το μικρότερο στοιχείο. Αν κάποιο από τα άλλα είναι μικρότερο, αποθηκεύουμε τη θέση του και το περιεχόμενό του ως "μικρότερο", και αυξάνουμε κατά ένα τον R5 για τα το συγκρίνουμε με το επόμενο.

Αφού βρούμε το μικρότερο αριθμό, αποθηκεύουμε σε έναν βοηθητικό καταχωρητή R5 τη τιμή του R7 ώστε να ανταλλάξουμε τις θέσεις μεταξύ της τωρινής και της μικρότερης τιμής. Στην συνέχεια αυξάνουμε τη θέση του πρώτου αριθμού κατά ένα, και επαναλαμβάνουμε την διαδικασία για όλα τα στοιχεία του πίνακα.

0x45	0x82	0x34	0xDA	0x10	0x28
0x10	0x82	0x34	0xDA	0x45	0x28
0x10	0x28	0x34	0xDA	0x45	0x82
0x10	0x28	0x34	0xDA	0x45	0x82
0x10	0x28	0x34	0x45	0xDA	0x82
0x10	0x28	0x34	0x45	0x82	0xDA

ii. Εκτέλεση αλγορίθμου και επιβεβαίωση ορθότητας αποτελεσμάτων

0	.arm	56		
1	.text	57	TestRoutine:	
2	.global main	58		
3		59	STMDB R13!, {R0-R12}	
4	main:	60		
5	STMDB R13!, {R0 - R12, R14}	61	LDR R0, =Table	Η TestRoutine χρησιμοποιεί τον R12
6		62	MOV R12, #1 @True	για να εξάγει το αποτέλεσμα του
7	LDR R0, =Table	63	MOV R11, #1	ελέγχου της ταξινόμησης
8	MOV R1, #20 @table_size	64		(1=True, 0=False).
9		65	Looptest:	
10	BL Insertion	66	LDRB R9, [R0], #1	Στον R9, R10 αποθηκεύονται οι δύο
11	BL TestRoutine	67	LDRB R10, [R0]	αριθμοί που κάθε φορά ελέγχονται. Αν
12		68		ο δεύτερος είναι μικρότερος,
13	LDMIA R13!, {R0- R12, R14}	69	CMP R1, R2	αποθηκεύεται το 0 στον R12.
14		70	MOVHI R12, #0 @False	
15	Insertion:	71		Η επανάληψη συνεχίζεται για όλους
16	STMDB R13!, {R0-R12}	72	ADD R11, R11, #0x01	τους 20 αριθμούς μέσω του μετρητή
17		73	CMP R11, #0x14	R11 και το αποτέλεσμα
18	LDRB R2, [R0, #0]	74	BNE Looptest	αποθηκεύεται στην μνήμη.
19	MOV R3, #0	75		
20	MOV R4, #0	76	STRB R12, [R0, #1]	
21	MOV R5, #1	77		
22		78	LDMIA R13!, {R0-R12}	
23	MainLoop:	79		
24		80	MOV PC, LR	
25	ADD R5, R3, #1	81		
26	LDRB R2, [R0, R3]	82		
27	MOV R4, R3	83		
28		84	.data	
29	Loop:	85		
30	LDRB R6, [R0, R5]	86	Table:	
31		87	.byte 0x12, 0x54, 0x4a, 0x02,	
32	CMP R2, R6	88	0x8a, 0xff	
33	MOVHI R4, R5	88	.byte 0x9f, 0x0f, 0xac, 0x78,	
34	MOVHI R2, R6	90	0x34, 0xda	
35		91	.byte 0x9c, 0x45, 0x99, 0xea,	
36	ADD R5, R5, #1	92	0x48, 0x05	
37	CMP R5, R1	93	.byte 0x76, 0x36, 0x00	
38		94		
39	BLT Loop	95		
40		96		
41	LDRB R7, [R0, R3]	97		
42	STRB R2, [R0, R3]	98		
43	STRB R7, [R0, R4]	99		
44	ADD R3, R3, #1			
45				
46				
47	SUB R8, R1, #1			
48				
49	CMP R3, R8			
50	BLT MainLoop			
51				
52	LDMIA R13!, {R0-R12}			
53	MOV PC, LR			
54				
55				