ΕΡΓΑΣΤΗΡΙΟ ΒΑΣΙΚΑ ΘΕΜΑΤΑ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Βασιλική Ευαγγελία Δούρου- Α.Μ.:1072633

Παύλος Πεσκελίδης- Α.Μ.:1072483

**ΕΡΓΑΣΙΑ 5:**

1. Ο κώδικας για την υπορουτίνα είναι ο ακόλουθος:

Subrtn:

STMDB R13!, {R2-R5}

BigLoop:

ADD R3, R0, #1

Loop:

LDRSB R2, [R0]

LDRSB R4, [R3]

CMP R2,R4

MOVGT R5, R2

MOVGT R2, R4

MOVGT R4, R5

STRB R2, [R0]

STRB R4, [R3]

ADD R3, R3, #1

CMP R3, R1

BLO Loop

ADD R0, R0, #1

CMP R0, R6

BLO BigLoop

LDMIA R13!, {R2-R5}

MOV PC,LR

Αρχικά, δεσμεύουμε τους καταχωρητές που πρόκειται να χρησιμοποιήσουμε στη διάρκεια της υπορουτίνας στη μνήμη.

Έπειτα, ξεκινάει μία δομή επανάληψης που λέγεται BigLoop, στη διάρκεια της οποίας ο καταχωρητής R3 παίρνει ως τιμή την επόμενη διεύθυνση από αυτή που δείχνει ο R0, γίνεται μία δεύτερη δομή επανάληψης που λέγεται Loop, στη διάρκεια της οποίας συγκρίνουμε το στοιχείο που περιέχει η διεύθυνση που δείχνει ο R0 με όλα τα υπόλοιπα στοιχεία του πίνακα, και τέλος αυξάνουμε κατά 1 το περιεχόμενο του R0 για να δείχνει στο επόμενο στοιχείο. Το BigLoop τερματίζεται όταν ο R0 πάρει ως τιμή τη διεύθυνση του τελευταίου στοιχείου του πίνακα και, στη συνέχεια, απελευθερώνονται οι καταχωρητές που είχαν δεσμευτεί για την υπορουτίνα και επιστρέφεται ο PC στο σημείο όπου είχε κληθεί η Subrtn. Στη διάρκεια του εμφωλευμένου Loop, φορτώνουμε στον καταχωρητή R2 το περιεχόμενο της διεύθυνσης που δείχνει ο R0 σε μορφή byte και στον R4 το περιεχόμενο της διεύθυνσης που δείχνει ο R3 σε μορφή byte. Έπειτα, συγκρίνουμε το R2 με το R4 και αν το R2 είναι μεγαλύτερο, μεταφέρουμε το περιεχόμενο του στον καταχωρητή R5, μετά το περιεχόμενο του R4 στον R2 και τέλος πάλι το περιεχόμενο του R5 στον R4, και μεταφέρουμε τα περιεχόμενα των R2,R4 στη διεύθυνση που δείχνουν οι R0, R3 αντίστοιχα, σε μορφή byte. Έτσι, σε περίπτωση που το περιεχόμενο της διεύθυνσης που δείχνει ο R0 είναι μεγαλύτερο από κάποιο επόμενο, θα γίνει ανταλλαγή των τιμών μεταξύ εκείνων των θέσεων. Τέλος, θα αυξηθεί κατά 1 το περιεχόμενο του R3 και θα συγκριθεί ο R3 με τον R1, ώστε να σταματήσει αυτή η δομή επανάληψης όταν το περιεχόμενο της διεύθυνσης που δείχνει ο R0 έχει συγκριθεί με όλο τον υπόλοιπο πίνακα.

1. Ολόκληρος ο κώδικας για την εκτέλεση του ζητούμενου για 20 τυχαίους αριθμούς είναι ο ακόλουθος:

.arm

.text

.global main

main:

STMDB R13!, {R0-R12, R14}

LDR R0, =Values

LDR R1, =Values + 20

ADD R6, R0, #19

BL Subrtn

LDMIA R13!, {R0-R12, PC}

Subrtn:

STMDB R13!, {R2-R5}

BigLoop:

ADD R3, R0, #1

Loop:

LDRSB R2, [R0]

LDRSB R4, [R3]

CMP R2,R4

MOVGT R5, R2

MOVGT R2, R4

MOVGT R4, R5

STRB R2, [R0]

STRB R4, [R3]

ADD R3, R3, #1

CMP R3, R1

BLO Loop

ADD R0, R0, #1

CMP R0, R6

BLO BigLoop

LDMIA R13!, {R2-R5}

MOV PC,LR

.data

Values:

.byte 0x01, 0x04, 0x07, 0x00, 0x0A, 0x03, 0x06, 0x08, 0x0B, 0x02, 0x05, 0x09, 0x0D, 0x0C, 0x10, 0x0F, 0x0E, 0x1A, 0x012, 0x14

Οι τυχαίοι αριθμοί που χρησιμοποιήθηκαν είναι οι εξής: 0x01, 0x04, 0x07, 0x00, 0x0A, 0x03, 0x06, 0x08, 0x0B, 0x02, 0x05, 0x09, 0x0D, 0x0C, 0x10, 0x0F, 0x0E, 0x1A, 0x012, 0x14.

Αρχικά, δεσμεύουμε τους καταχωρητές που πρόκειται να χρησιμοποιήσουμε στη μνήμη και αρχικοποιούμε την τιμή του R0 με τη διεύθυνση της ετικέτας Values, για να δείχνει στο πρώτο στοιχείο του πίνακα, την τιμή του R1 με τη διεύθυνση της Values+20 για να δείχνει στο τελευταίο στοιχείο του πίνακα και, και την τιμή του R6 με R0 + 19 για να δείχνει στο προτελευταίο στοιχείο του πίνακα, για να μην μπει το BigLoop της υπορουτίνας σε μία επιπλέον επανάληψη. Τέλος, καλούμε την υπορουτίνα και μετά την κλήση της αποδεσμεύουμε τους καταχωρητές που χρησιμοποιήσαμε.