DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

EMBEDDED SYSTEMS

Experiment 9

Submitted to,

Dr. Shireesh Kumar Rai
Asst. Professor

Submitted by,

T Raja Aadhithan
602162021
M.Tech (VLSI Design)

Experiment 9

Aim:

To write an ARM Assembly Language to arrange the numbers in ascending and descending order.

Tool Used:

Keil uVision4

Theory:

The numbers in the memory location are repeatedly compared in iterative manner and using the carry flag's result they are restored in either ascending or descending order.

Code:

Ascending:

```
 AREA PROGRAM, CODE, READONLY
 ENTRY
MAIN
        MOV R0, #9 ; n-1 memory loctions
LOOP1   LDR R1, =0X1000 ; starting memory location
        ADD R2,R1,#1 ; the 2nd number location
        MOV R3,R0 ; copy the value
LOOP2   LDRB R4, [R1], #1 ; load 1st number
        LDRB R5, [R2], #1 ; load 2nd number
        CMP R4,R5 ; compare both  numbers
        STRCSB R4, [R2,#-1] ; swap is greater
        STRCSB R5, [R1,#-1] ; r4 is expected to be lesser
        SUBS R3,R3,#1 ; decrement counter
        BNE LOOP2
        SUBS R0,R0,#1 ; decrement counter
        BNE LOOP1
    END
```

Decending:

```
AREA PROGRAM, CODE, READONLY
 ENTRY
MAIN
        MOV R0, #9 ; n-1 memory loctions
LOOP1   LDR R1, =0X1000 ; starting memory location
        ADD R2,R1,#1 ; the 2nd number location
```

```
        MOV R3,R0 ; copy the value
LOOP2   LDRB R4, [R1], #1 ; load 1st number
        LDRB R5, [R2], #1 ; load 2nd number
        CMP R4,R5 ; compare both  numbers
        STRCCB R4, [R2,#-1] ; swap is lesser
        STRCCB R5, [R1,#-1] ; r4 is expected to be greater
        SUBS R3,R3,#1 ; decrement counter
        BNE LOOP2
        SUBS R0,R0,#1 ; decrement counter
        BNE LOOP1
    END
```

Output:

```
Running with Code Size Limit: 32K
Load "C:\\Users\\User\\Documents\\Code-sync\\Keil\\ARM\\Experiment 9\\ecxp9.axf"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 104 Bytes (0%)
```

Input given before running:

```
0x00001000: 10 54 22 03 00 11 10 FF 21 01 00 00 00 00 00
0x0000100F: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Output after Ascending:

```
0x00001000: 00 01 03 10 10 11 21 22 53 FF 00 00 00 00 00
0x0000100F: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Output after Descending:

```
0x00001000: FF 54 22 21 11 10 10 03 01 00 00 00 00 00 00
0x0000100F: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Result:

The experiments on division operation has been performed and verified to be correct.