

## Experiment 5

### Aim:

To write an ARM Assembly Language to find the number of bytes in a set of 10 locations that match the value 0xAC

### Tool Used:

Keil uVision4

### Theory:

LDRB is used to copy just 1 Byte of data to the lower location of the memory. LDR load multiple register locations with starting address mentioned. CMP compares two operands and if zero sets the zero flag. The EQ condition checks the zero flag for set to let the process happen.

### Code:

```
AREA PROGRAM, CODE, READONLY
ENTRY
MAIN
    MOV R0, #10
    LDR R1, =0x00001000
LOOP
    LDRB R2, [R1], #1
    CMP R2, #0xAC
    ADDEQ R3,R3,#1
    SUBS R0,R0,#1
    BNE LOOP
END
```

### Memory content

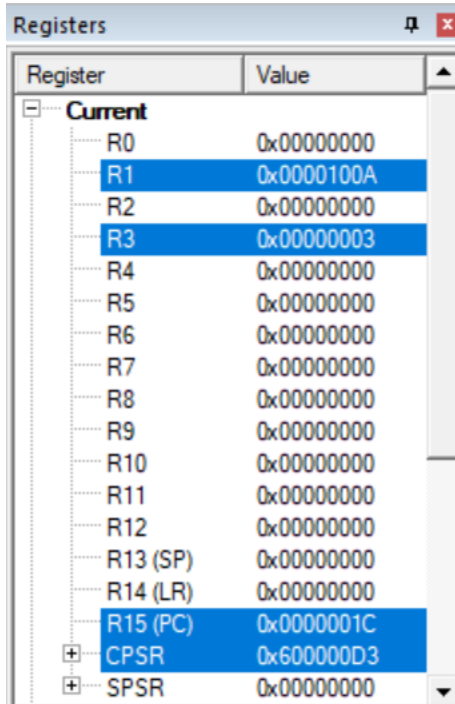
Memory 1															
Address: 0x00001000															
0x00001000:	00	AC	00	AC	AC	00	00	00	00	00	00	00	00	00	00
0x00001015:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0000102A:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x0000103F:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0x00001054:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

```
Command
*** Error: 'C:\Keil\ARM\BIN\DARMO.DLL' not found
Running with Code Size Limit: 32K
Load "C:\\Users\\singh\\Documents\\keil embedded system\\experiment 5\\exp5.a

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 28 Bytes (0%)

*** error 65: access violation at 0x0000001C : no 'execute/read' permission
```

## Register Contents



A screenshot of a 'Registers' window from a debugger. The window has a title bar with a pin icon and a close button. It contains a table with two columns: 'Register' and 'Value'. The table lists registers R0 through R15, CPSR, and SPSR. R0, R2, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13 (SP), R14 (LR), and SPSR have values of 0x00000000. R1 has a value of 0x0000100A. R3 has a value of 0x00000003. R15 (PC) has a value of 0x0000001C. CPSR has a value of 0x600000D3. The registers R1, R3, R15 (PC), and CPSR are highlighted in blue. There are expand/collapse icons to the left of the register names.

Register	Value
<b>Current</b>	
R0	0x00000000
R1	0x0000100A
R2	0x00000000
R3	0x00000003
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000001C
CPSR	0x600000D3
SPSR	0x00000000

### Result:

The experiments on compare operation have been performed and verified to be correct.