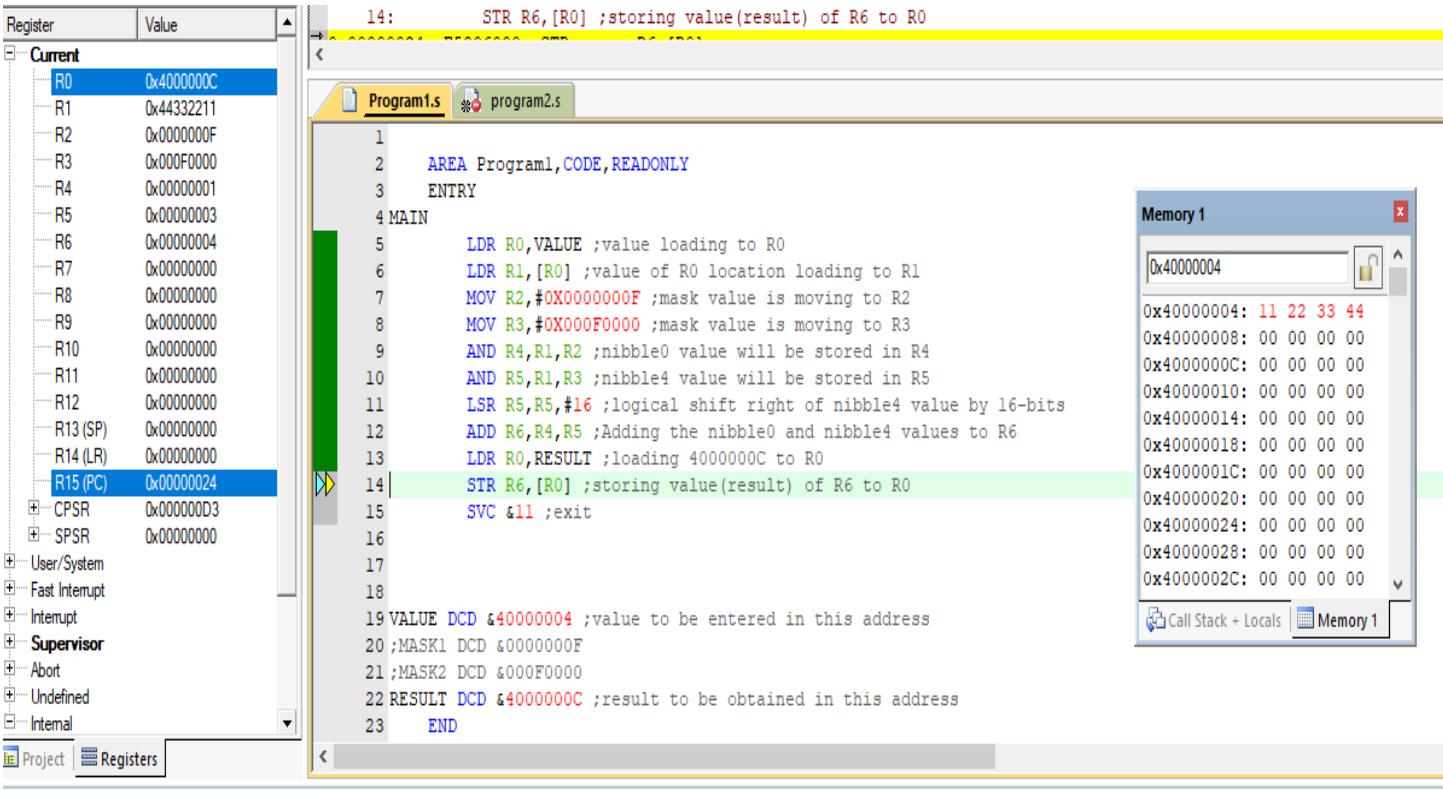
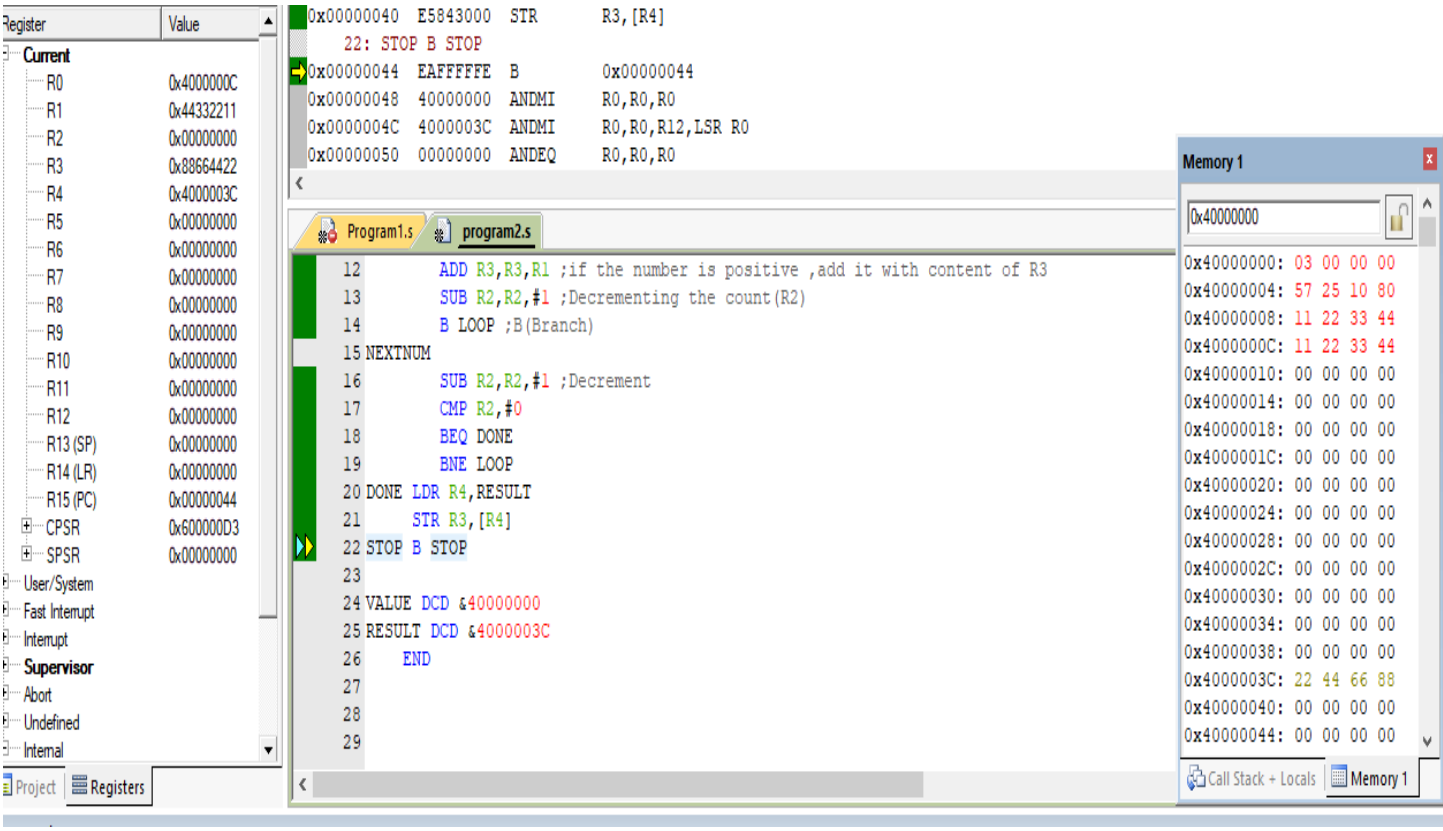
1) Assume a 32-bit number in 40000004h. Add nibble 4 and nibble0 & store the result in 4000000Ch.

```
        AREA Program1,CODE,READONLY
    ENTRY
MAIN
        LDR R0,VALUE ;value loading to R0
        LDR R1,[R0] ;value of R0 location loading to R1
        MOV R2,#0X0000000F ;mask value is moving to R2
        MOV R3,#0X000F0000 ;mask value is moving to R3
        AND R4,R1,R2 ;nibble0 value will be stored in R4
        AND R5,R1,R3 ;nibble4 value will be stored in R5
        LSR R5,R5,#16 ;logical shift right of nibble4 value by 16-bits
        ADD R6,R4,R5 ;Adding the nibble0 and nibble4 values to R6
        LDR R0,RESULT ;loading 4000000C to R0
        STR R6,[R0] ;storing value(result) of R6 to R0
        SVC &11 ;exit

VALUE DCD &40000004 ;value to be entered in this address
RESULT DCD &4000000C ;result to be obtained in this address
    END
```


Register
Value
Current
R0 0x4000000C
R1 0x44332211
R2 0x0000000F
R3 0x000F0000
R4 0x00000001
R5 0x00000003
R6 0x00000004
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000024
CPSR 0x000000D3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
Project
Registers
14: STR R6,[R0] ;storing value(result) of R6 to R0
Program1.s
program2.s
1
2 AREA Program1,CODE,READONLY
3 ENTRY
4 MAIN
5 LDR R0,VALUE ;value loading to R0
6 LDR R1,[R0] ;value of R0 location loading to R1
7 MOV R2,#0X0000000F ;mask value is moving to R2
8 MOV R3,#0X000F0000 ;mask value is moving to R3
9 AND R4,R1,R2 ;nibble0 value will be stored in R4
10 AND R5,R1,R3 ;nibble4 value will be stored in R5
11 LSR R5,R5,#16 ;logical shift right of nibble4 value by 16-bits
12 ADD R6,R4,R5 ;Adding the nibble0 and nibble4 values to R6
13 LDR R0,RESULT ;loading 4000000C to R0
14 STR R6,[R0] ;storing value(result) of R6 to R0
15 SVC &11 ;exit
16
17
18
19 VALUE DCD &40000004 ;value to be entered in this address
20 ;MASK1 DCD &0000000F
21 ;MASK2 DCD &000F0000
22 RESULT DCD &4000000C ;result to be obtained in this address
23 END
Memory 1
0x40000004
0x40000004: 11 22 33 44
0x40000008: 00 00 00 00
0x4000000C: 00 00 00 00
0x40000010: 00 00 00 00
0x40000014: 00 00 00 00
0x40000018: 00 00 00 00
0x4000001C: 00 00 00 00
0x40000020: 00 00 00 00
0x40000024: 00 00 00 00
0x40000028: 00 00 00 00
0x4000002C: 00 00 00 00
Call Stack + Locals
Memory 1

2. Consider an array of number present from 40000000H. Add only if the numbers are positive. 40000000H has the count of the array.

```
        AREA program2,CODE,READONLY
   ENTRY
MAIN
    LDR R0,VALUE ; value loading to R0
    LDR R2,[R0];value of R0 location loading to R2(count)
    EOR R3,R3,R3 ;EXOR operation to clear the register and hold the sum
LOOP   CMP R2,#0 ;compare R2(count) with zero
    BEQ DONE ;Branch if Equal to zero
    LDR R1,[R0,#4]! ;load the array elements to R1
    CMP R1,#0 ;compare the content of Register R1 with 0
    BMI NEXTNUM ;BMI(Branch if Minus) checks the negative flag , if it sets the branch to loop
    ADD R3,R3,R1 ;if the number is positive ,add it with content of R3
    SUB R2,R2,#1 ;Decrementing the count(R2)
    B LOOP ;B(Branch)
NEXTNUM
    SUB R2,R2,#1 ;Decrement
    CMP R2,#0
    BEQ DONE
    BNE LOOP
DONE LDR R4,RESULT
   STR R3,[R4]
STOP B STOP

VALUE DCD &40000000
RESULT DCD &4000003C
   END
```


Register
Value
Current
R0 0x4000000C
R1 0x44332211
R2 0x00000000
R3 0x88664422
R4 0x4000003C
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000044
CPSR 0x600000D3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
Project
Registers
0x00000040 E5843000 STR R3,[R4]
22: STOP B STOP
0x00000044 EAFFFFFE B 0x00000044
0x00000048 40000000 ANDMI R0,R0,R0
0x0000004C 4000003C ANDMI R0,R0,R12,LSR R0
0x00000050 00000000 ANDEQ R0,R0,R0
Program1.s
program2.s
12 ADD R3,R3,R1 ;if the number is positive ,add it with content of R3
13 SUB R2,R2,#1 ;Decrementing the count(R2)
14 B LOOP ;B(Branch)
15 NEXTNUM
16 SUB R2,R2,#1 ;Decrement
17 CMP R2,#0
18 BEQ DONE
19 BNE LOOP
20 DONE LDR R4,RESULT
21 STR R3,[R4]
22 STOP B STOP
23
24 VALUE DCD &40000000
25 RESULT DCD &4000003C
26 END
27
28
29
Memory 1
0x40000000
0x40000000: 03 00 00 00
0x40000004: 57 25 10 80
0x40000008: 11 22 33 44
0x4000000C: 11 22 33 44
0x40000010: 00 00 00 00
0x40000014: 00 00 00 00
0x40000018: 00 00 00 00
0x4000001C: 00 00 00 00
0x40000020: 00 00 00 00
0x40000024: 00 00 00 00
0x40000028: 00 00 00 00
0x4000002C: 00 00 00 00
0x40000030: 00 00 00 00
0x40000034: 00 00 00 00
0x40000038: 00 00 00 00
0x4000003C: 22 44 66 88
0x40000040: 00 00 00 00
0x40000044: 00 00 00 00
Call Stack + Locals
Memory 1