NAME :- SHREYA GIRISH

REGISTRATION NUMBER - 211039020

Q1) Assume a 32-bit number in 40000004H. Add nibble4 and nibble0 and store the result in 4000000CH.

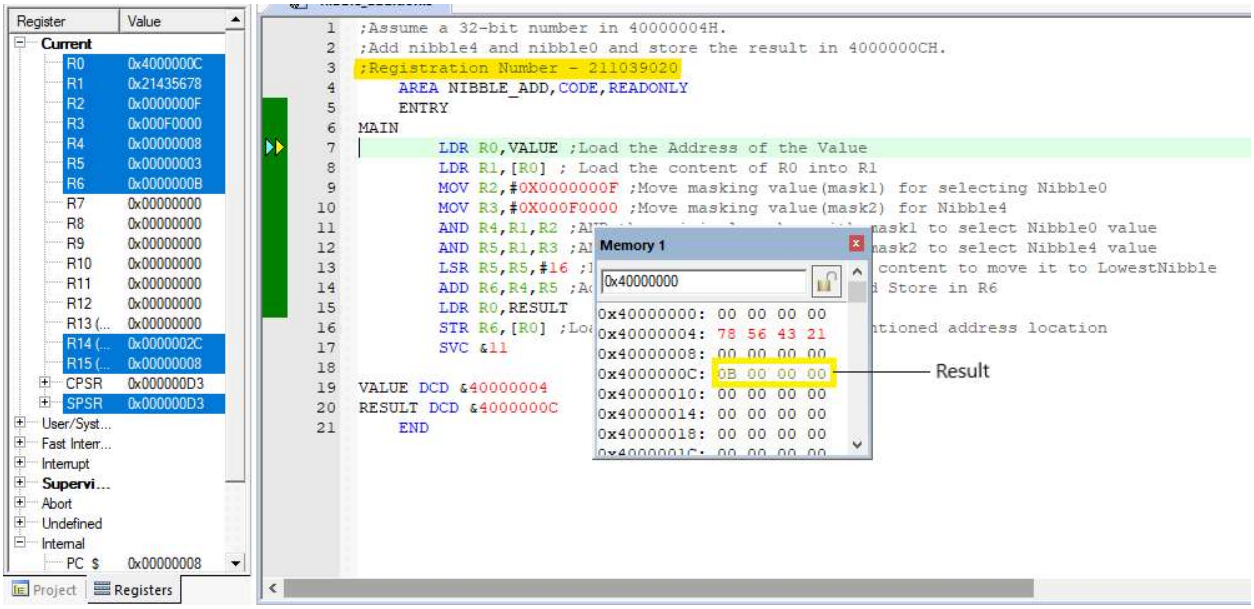
SOURCE CODE:-

```
;Assume a 32-bit number in 40000004H.
;Add nibble4 and nibble0 and store the result in 4000000CH.
        AREA NIBBLE_ADD,CODE,READONLY
        ENTRY
MAIN
                LDR R0, VALUE ; Load the Address of the Value
                LDR R1, [R0] ; Load the content of R0 into R1
                MOV R2, #0X0000000F ; Move masking value(0x0000000F) for selecting
;Nibble0
                MOV R3, #0X000F0000 ; Move masking value(0x000F0000) for selecting
;Nibble4
                AND R4, R1, R2 ; AND the original number with mask1 to select Nibble0 value
                AND R5, R1, R3 ; AND the original number with mask2 to select Nibble4 value
                LSR R,R5,#16 ; LogicalShiftLeft R5 register content to move it to LowestNibble
                ADD R6, R4, R5 ; Add both the nibble values and Store in R6
                LDR R0,RESULT
                STR R6,[R0] ; Load the value of result to mentioned address location
                SVC &11
VALUE DCD &40000004
RESULT DCD &4000000C
        END
```

```
;Assume a 32-bit number in 40000004H.
;Add nibble4 and nibble0 and store the result in 4000000CH.
;Registration Number - 211039020
    AREA NIBBLE_ADD,CODE,READONLY
    ENTRY
MAIN
        LDR R0,VALUE        ;Load the Address of the Value
        LDR R1,[R0]         ; Load the content of R0 into R1
        MOV R2,#0X0000000F  ;Move masking value(0x0000000F) for selecting Nibble0
        MOV R3,#0X000F0000  ;Move masking value(0x000F0000) for selecting Nibble4
        AND R4,R1,R2        ;AND the original number with mask1 to select Nibble0 value
        AND R5,R1,R3        ;AND the original number with mask2 to select Nibble4 value
        LSR R5,R5,#16       ;LogicalShiftLeft R5 register content to move it to LowestNibble
        ADD R6,R4,R5        ;Add both the nibble values and Store in R6
        LDR R0,RESULT
        STR R6,[R0]         ;Load the value of result to mentioned address location
        SVC &11

VALUE DCD &40000004
RESULT DCD &4000000C
    END
```

OUTPUT:-



Q2) Consider an array of number present from 40000000H. Add only if the numbers are positive. 40000000H has the count of the array.

SOURCE CODE:-

```
;Consider an array of number present from 40000000H. Add only if the numbers ;are positive.
;40000000H has the count of the array.
;A number is said to be negative if its MSB bit is set to 1.
;Registration Number - 211039020
                AREA ADD_ARRAY,CODE,READONLY
                ENTRY
MAIN
                LDR R0,TABLE; In the table first value is count followed by array elements
                LDR R2,[R0];Load the count
                EOR R3,R3,R3;Clear Register R3 for storing sum of positive elements
LOOP
                  CMP R2,#0 ; Compare count with Zero
BEQ DONE
                  LDR R1,[R0,#4]! ;Load the array elements to R1 register
                  CMP R1,#0 ;Compare R1 register content with 0
                  BMI LOOP1 ;If the number is negative then Branch to Loop1 and
;decrement the count
;BMI - Branch if minus ; It checks the Negative Flag if it is set Branch to Loop1
                  ADD R3,R3,R1 ;If the number is positive Add it with R3 register content
                  SUB R2,R2,#1 ;Decrement the count
                  B LOOP

LOOP1
                        SUB R2,R2,#1
                        CMP R2,#0
```

BEQ DONE

BNE LOOP

DONE LDR R4,RESULT

STR R3,[R4] ;Load the Result in R3 to desired location

STOP B STOP

TABLE DCD &40000000

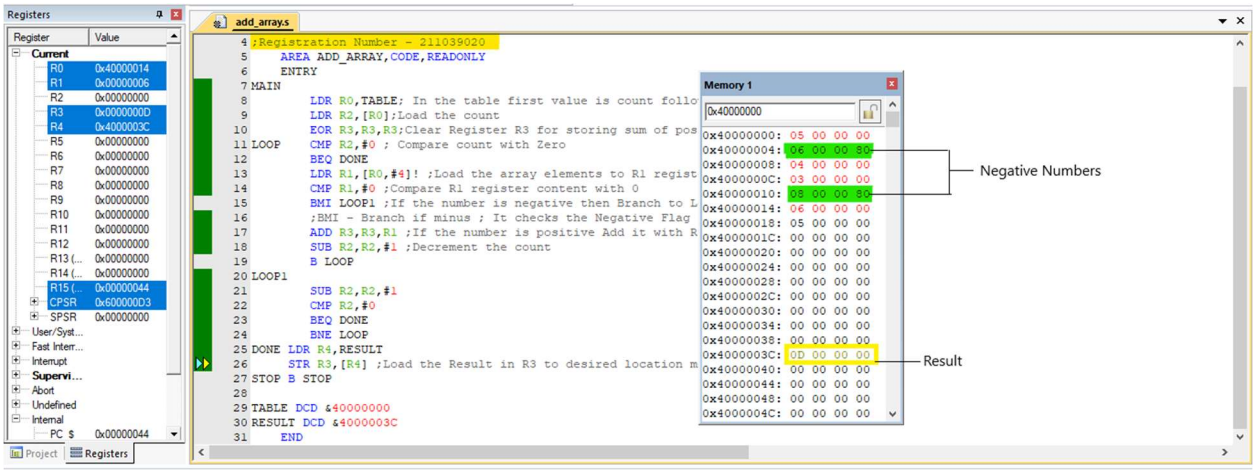
RESULT DCD &4000003C

END

```
;Consider an array of number present from 40000000H. Add only if the numbers are positive.
;40000000H has the count of the array.
;A number is said to be negative if its MSB bit is set to 1.
;Registration Number - 211039020
    AREA ADD_ARRAY,CODE,READONLY
    ENTRY
MAIN
        LDR R0,TABLE    ; In the table first value is count followed by array elements
        LDR R2,[R0]     ;Load the count
        EOR R3,R3,R3    ;Clear Register R3 for storing sum of positive elements
LOOP    CMP R2,#0       ; Compare count with Zero
        BEQ DONE
        LDR R1,[R0,#4]! ;Load the array elements to R1 register
        CMP R1,#0       ;Compare R1 register content with 0
        BMI LOOP1       ;If the number is negative then Branch to Loop1 and decrement the count
                        ;BMI - Branch if minus
                        ; It checks the Negative Flag if it is set Branch to Loop1
        ADD R3,R3,R1    ;If the number is positive Add it with R3 register content
        SUB R2,R2,#1    ;Decrement the count
        B LOOP
LOOP1
        SUB R2,R2,#1
        CMP R2,#0
        BEQ DONE
        BNE LOOP
DONE LDR R4,RESULT
     STR R3,[R4]        ;Load the Result in R3 to desired location mentioned below
STOP B STOP

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

OUTPUT:-


Registers
Register
Value
Current
R0 0x40000014
R1 0x00000006
R2 0x00000000
R3 0x0000000D
R4 0x4000003C
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (... 0x00000000
R14 (... 0x00000000
R15 (... 0x00000044
CPSR 0x600000D3
SPSR 0x00000000
User/Syst...
Fast Interr...
Interrupt
Supervi...
Abort
Undefined
Internal
PC $ 0x00000044
Project
Registers
add_array.s
;Registration Number - 211039020
AREA ADD_ARRAY,CODE,READONLY
ENTRY
MAIN
LDR R0,TABLE; In the table first value is count follo
LDR R2,[R0];Load the count
EOR R3,R3,R3;Clear Register R3 for storing sum of pos
LOOP CMP R2,#0 ; Compare count with Zero
BEQ DONE
LDR R1,[R0,#4]! ;Load the array elements to R1 regist
CMP R1,#0 ;Compare R1 register content with 0
BMI LOOP1 ;If the number is negative then Branch to L
;BMI - Branch if minus ; It checks the Negative Flag
ADD R3,R3,R1 ;If the number is positive Add it with R
SUB R2,R2,#1 ;Decrement the count
B LOOP
LOOP1
SUB R2,R2,#1
CMP R2,#0
BEQ DONE
BNE LOOP
DONE LDR R4,RESULT
STR R3,[R4] ;Load the Result in R3 to desired location m
STOP B STOP
TABLE DCD &40000000
RESULT DCD &4000003C
END
Memory 1
0x40000000
0x40000000: 05 00 00 00
0x40000004: 06 00 00 80
0x40000008: 04 00 00 00
0x4000000C: 03 00 00 00
0x40000010: 08 00 00 80
0x40000014: 06 00 00 00
0x40000018: 05 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: 0D 00 00 00
0x40000040: 00 00 00 00
0x40000044: 00 00 00 00
0x40000048: 00 00 00 00
0x4000004C: 00 00 00 00
Negative Numbers
Result