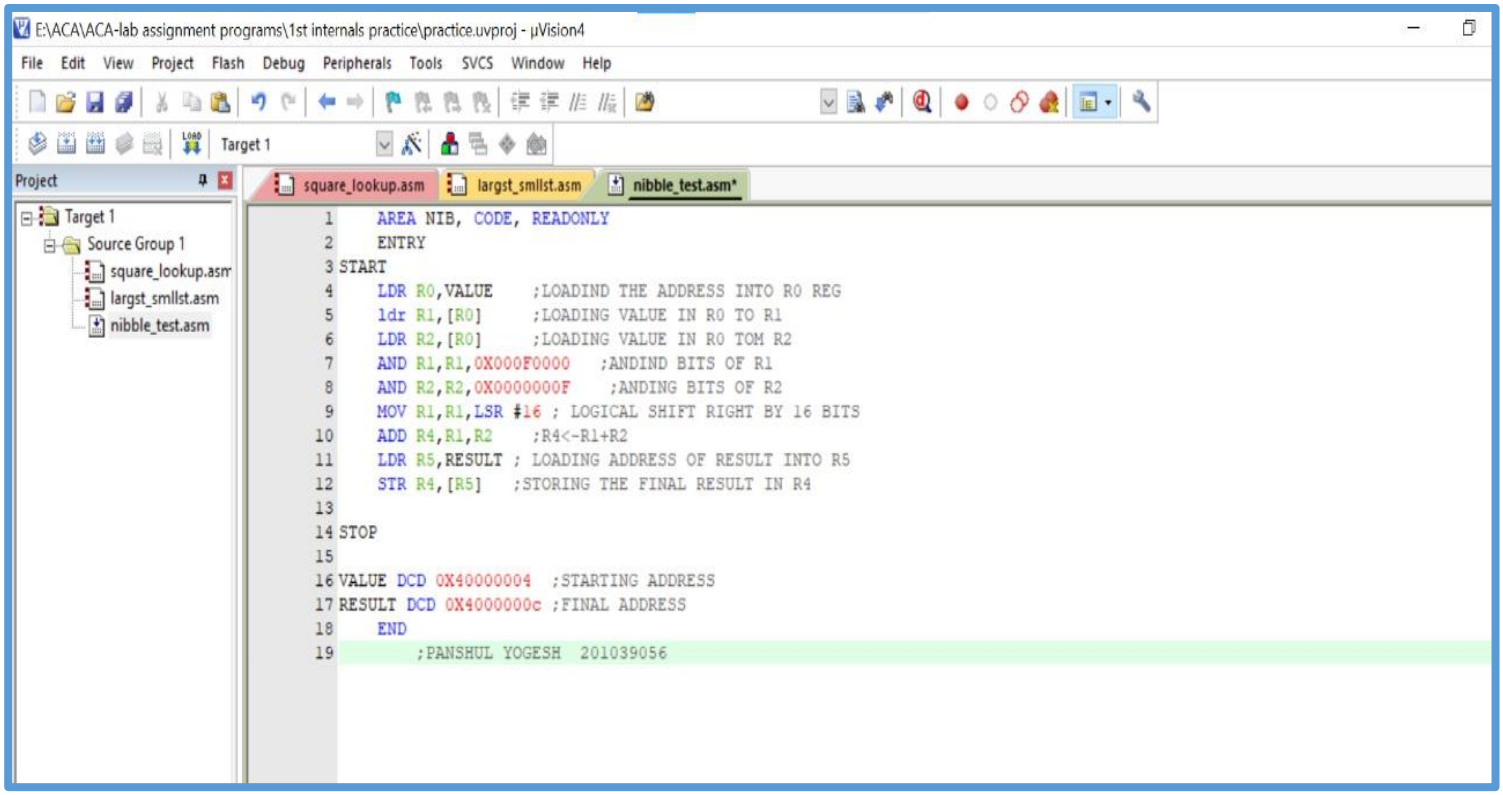
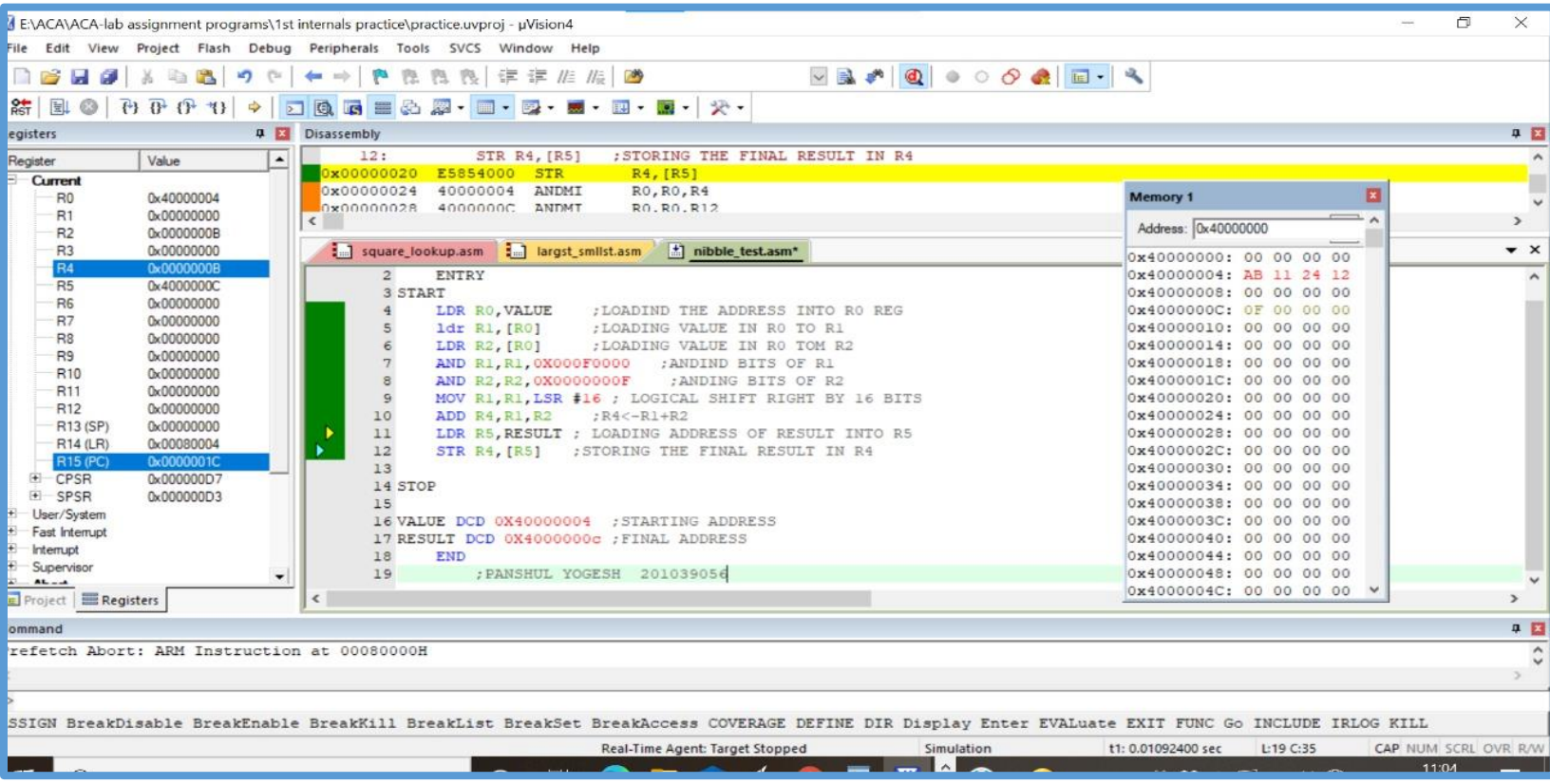
Q1.

CODE


E:\ACA\ACA-lab assignment programs\1st internals practice\practice.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Target 1
Source Group 1
square_lookup.asm
largst_smllst.asm
nibble_test.asm
square_lookup.asm
largst_smllst.asm
nibble_test.asm*
1 AREA NIB, CODE, READONLY
2 ENTRY
3 START
4 LDR R0,VALUE ;LOADIND THE ADDRESS INTO R0 REG
5 ldr R1,[R0] ;LOADING VALUE IN R0 TO R1
6 LDR R2,[R0] ;LOADING VALUE IN R0 TOM R2
7 AND R1,R1,0X000F0000 ;ANDIND BITS OF R1
8 AND R2,R2,0X0000000F ;ANDING BITS OF R2
9 MOV R1,R1,LSR #16 ; LOGICAL SHIFT RIGHT BY 16 BITS
10 ADD R4,R1,R2 ;R4<-R1+R2
11 LDR R5,RESULT ; LOADING ADDRESS OF RESULT INTO R5
12 STR R4,[R5] ;STORING THE FINAL RESULT IN R4
13
14 STOP
15
16 VALUE DCD 0X40000004 ;STARTING ADDRESS
17 RESULT DCD 0X4000000c ;FINAL ADDRESS
18 END
19 ;PANSHUL YOGESH 201039056

OUTPUT


E:\ACA\ACA-lab assignment programs\1st internals practice\practice.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Register
Value
Current
R0 0x40000004
R1 0x00000000
R2 0x0000000B
R3 0x00000000
R4 0x0000000B
R5 0x4000000C
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00080004
R15 (PC) 0x0000001C
CPSR 0x000000D7
SPSR 0x000000D3
User/System
Fast Interrupt
Interrupt
Supervisor
Project
Registers
Disassembly
12: STR R4,[R5] ;STORING THE FINAL RESULT IN R4
0x00000020 E5854000 STR R4,[R5]
0x00000024 40000004 ANDMI R0,R0,R4
square_lookup.asm
largst_smllst.asm
nibble_test.asm*
2 ENTRY
3 START
4 LDR R0,VALUE ;LOADIND THE ADDRESS INTO R0 REG
5 ldr R1,[R0] ;LOADING VALUE IN R0 TO R1
6 LDR R2,[R0] ;LOADING VALUE IN R0 TOM R2
7 AND R1,R1,0X000F0000 ;ANDIND BITS OF R1
8 AND R2,R2,0X0000000F ;ANDING BITS OF R2
9 MOV R1,R1,LSR #16 ; LOGICAL SHIFT RIGHT BY 16 BITS
10 ADD R4,R1,R2 ;R4<-R1+R2
11 LDR R5,RESULT ; LOADING ADDRESS OF RESULT INTO R5
12 STR R4,[R5] ;STORING THE FINAL RESULT IN R4
13
14 STOP
15
16 VALUE DCD 0X40000004 ;STARTING ADDRESS
17 RESULT DCD 0X4000000c ;FINAL ADDRESS
18 END
19 ;PANSHUL YOGESH 201039056
Memory 1
Address: 0x40000000
0x40000000: 00 00 00 00
0x40000004: AB 11 24 12
0x40000008: 00 00 00 00
0x4000000C: 0F 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
0x40000030: 00 00 00 00
0x40000034: 00 00 00 00
0x40000038: 00 00 00 00
0x4000003C: 00 00 00 00
0x40000040: 00 00 00 00
0x40000044: 00 00 00 00
0x40000048: 00 00 00 00
0x4000004C: 00 00 00 00
Command
Prefetch Abort: ARM Instruction at 00080000H
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter EVALuate EXIT FUNC Go INCLUDE IRLOG KILL
Real-Time Agent: Target Stopped
Simulation
t1: 0.01092400 sec
L:19 C:35
CAP NUM SCRL OVR R/W

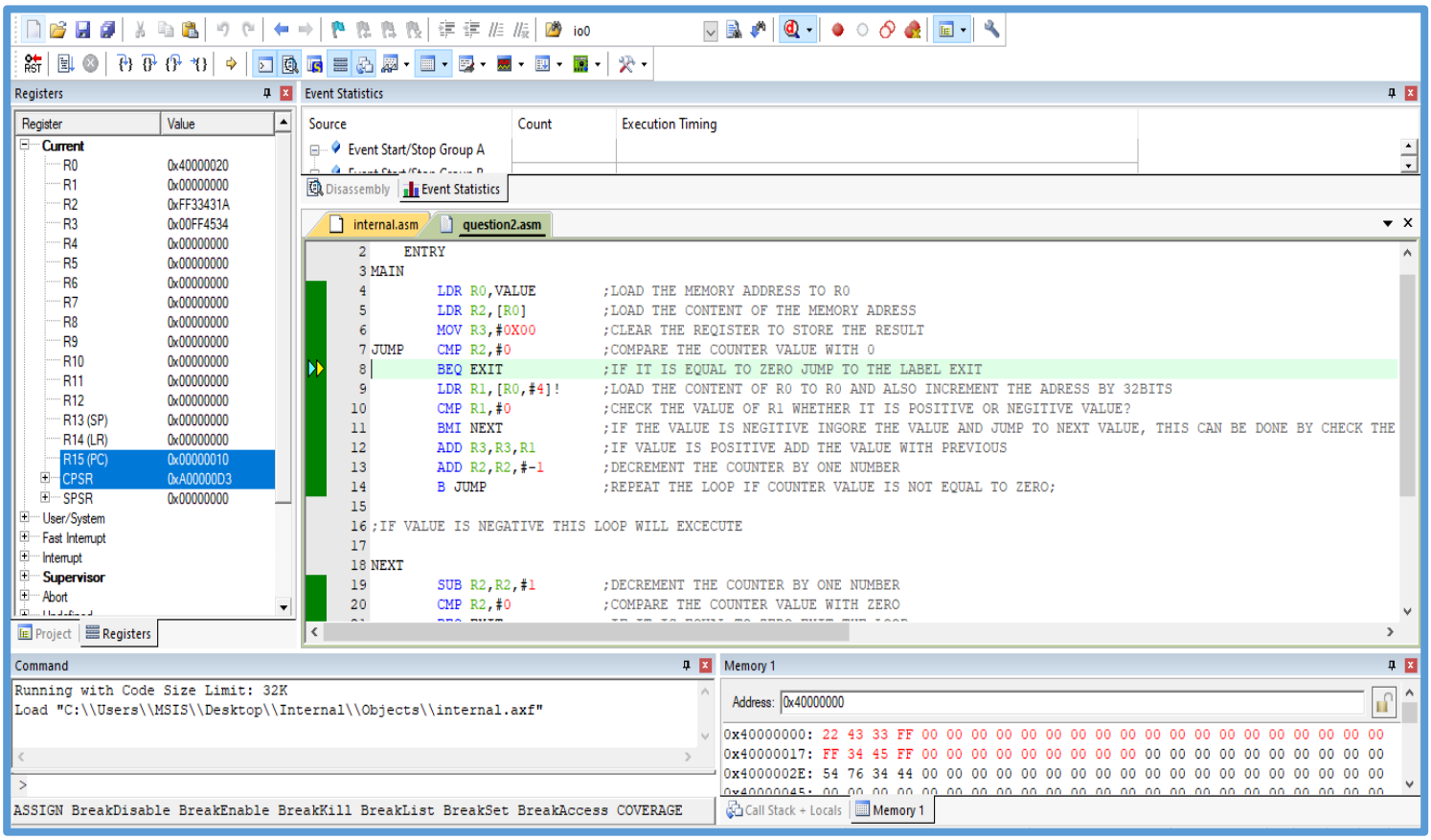
Q2.

CODE

question2.asm

```
   AREA ADD_ARRAY,CODE,READONLY
    ENTRY
MAIN
        LDR R0,VALUE        ;LOAD THE MEMORY ADDRESS TO R0
        LDR R2,[R0]         ;LOAD THE CONTENT OF THE MEMORY ADRESS
        MOV R3,#0X00        ;CLEAR THE REQISTER TO STORE THE RESULT
JUMP    CMP R2,#0           ;COMPARE THE COUNTER VALUE WITH 0
        BEQ EXIT            ;IF IT IS EQUAL TO ZERO JUMP TO THE LABEL EXIT
        LDR R1,[R0,#4]!     ;LOAD THE CONTENT OF R0 TO R0 AND ALSO INCREMENT THE ADRESS BY 32BITS
        CMP R1,#0           ;CHECK THE VALUE OF R1 WHETHER IT IS POSITIVE OR NEGITIVE VALUE?
        BMI NEXT            ;IF THE VALUE IS NEGITIVE INGORE THE VALUE AND JUMP TO NEXT VALUE, THIS CAN BE DONE BY CHECK THE N FLAG IN CPSR.
        ADD R3,R3,R1        ;IF VALUE IS POSITIVE ADD THE VALUE WITH PREVIOUS
        ADD R2,R2,#-1       ;DECREMENT THE COUNTER BY ONE NUMBER
        B JUMP              ;REPEAT THE LOOP IF COUNTER VALUE IS NOT EQUAL TO ZERO;

;IF VALUE IS NEGATIVE THIS LOOP WILL EXCECUTE

NEXT
        SUB R2,R2,#1        ;DECREMENT THE COUNTER BY ONE NUMBER
        CMP R2,#0           ;COMPARE THE COUNTER VALUE WITH ZERO
        BEQ EXIT            ;IF IT IS EQUAL TO ZERO EXIT THE LOOP
        BNE JUMP            ;IF COUNTER VALUE IS NOT EQUAL TO ZERO JUMP BACK AND REPEAT THE PROCESS
EXIT LDR R4,RESULT          ;LOAD 40000000 TO R4
     STR R3,[R4]            ;SOTRE THE RESULT IN MEMORY ADDRESS 40000000
STOP B STOP                 ;TERMINATION OF THE PROCESS

VALUE DCD &40000000         ;ASSIGNING 40000000 TO VARIABLE VALUE AND ACT AS A POINTER
RESULT DCD &4000003C        ;ASSIGNING 40000003 TO VARIABLE VALUE
    END                     ;END OF THE CODE

        ;PANSHUL YOGESH 201039056
```

OUTPUT


Registers
Register
Value
Current
R0 0x40000020
R1 0x00000000
R2 0xFF33431A
R3 0x00FF4534
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) 0x00000010
CPSR 0xA00000D3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Project
Registers
Event Statistics
Source
Count
Execution Timing
Event Start/Stop Group A
Disassembly
Event Statistics
internal.asm
question2.asm
2 ENTRY
3 MAIN
4 LDR R0,VALUE ;LOAD THE MEMORY ADDRESS TO R0
5 LDR R2,[R0] ;LOAD THE CONTENT OF THE MEMORY ADRESS
6 MOV R3,#0X00 ;CLEAR THE REQISTER TO STORE THE RESULT
7 JUMP CMP R2,#0 ;COMPARE THE COUNTER VALUE WITH 0
8 BEQ EXIT ;IF IT IS EQUAL TO ZERO JUMP TO THE LABEL EXIT
9 LDR R1,[R0,#4]! ;LOAD THE CONTENT OF R0 TO R0 AND ALSO INCREMENT THE ADRESS BY 32BITS
10 CMP R1,#0 ;CHECK THE VALUE OF R1 WHETHER IT IS POSITIVE OR NEGITIVE VALUE?
11 BMI NEXT ;IF THE VALUE IS NEGITIVE INGORE THE VALUE AND JUMP TO NEXT VALUE, THIS CAN BE DONE BY CHECK THE
12 ADD R3,R3,R1 ;IF VALUE IS POSITIVE ADD THE VALUE WITH PREVIOUS
13 ADD R2,R2,#-1 ;DECREMENT THE COUNTER BY ONE NUMBER
14 B JUMP ;REPEAT THE LOOP IF COUNTER VALUE IS NOT EQUAL TO ZERO;
15
16 ;IF VALUE IS NEGATIVE THIS LOOP WILL EXCECUTE
17
18 NEXT
19 SUB R2,R2,#1 ;DECREMENT THE COUNTER BY ONE NUMBER
20 CMP R2,#0 ;COMPARE THE COUNTER VALUE WITH ZERO
Command
Running with Code Size Limit: 32K
Load "C:\\Users\\MSIS\\Desktop\\Internal\\Objects\\internal.axf"
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE
Memory 1
Address: 0x40000000
0x40000000: 22 43 33 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000017: FF 34 45 FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000002E: 54 76 34 44 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1