MICROPROCESSOR BASED SYSTEMS DESIGN

UCS617

LAB ASSIGNMENT

ARM MICROPROCESSOR

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

Submitted By:

Rishabh Kumar 101611046
Rohan Subedi 101603284

Submitted To:

Ms. Swadha Gupta

INDEX

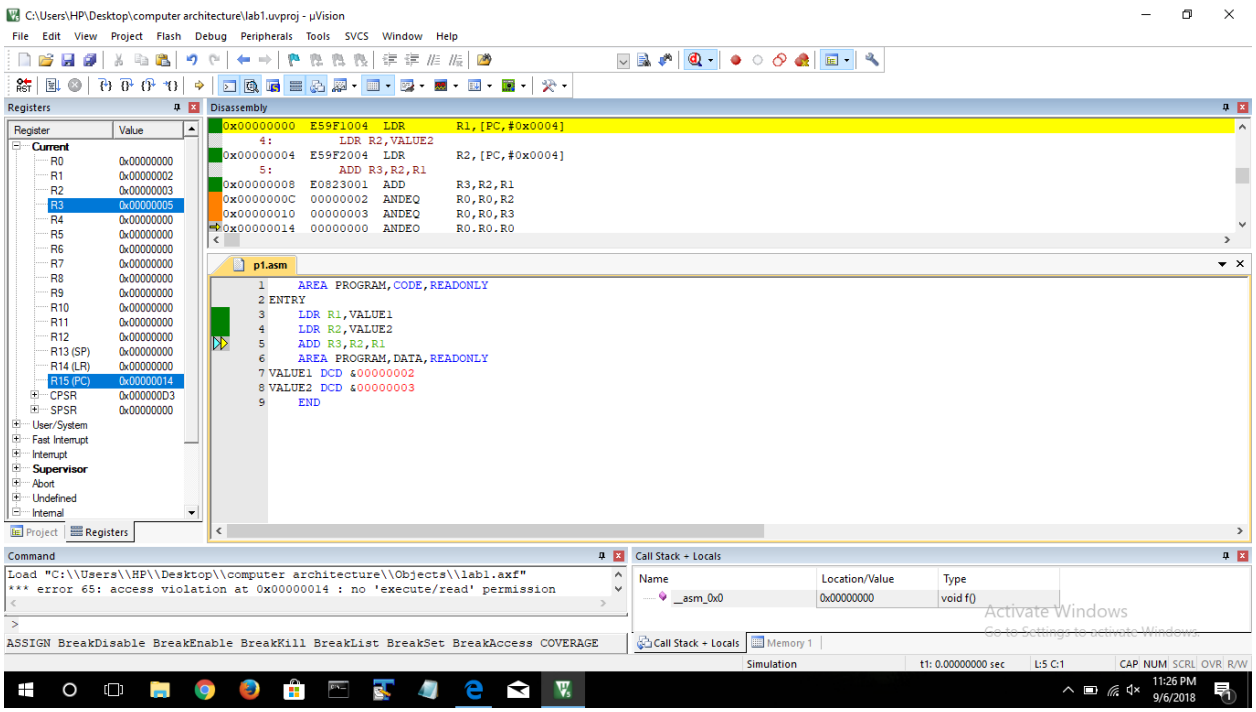
1. Write a program in ARM assembly language to add and subtract two 32-bit numbers using:
i) Direct addressing mode
ii) Indirect addressing mode
iii) Barrel shifter

ANS:

ADDITION OF TWO NUMBERS USING DIRECT ADDRESSING

CODE:

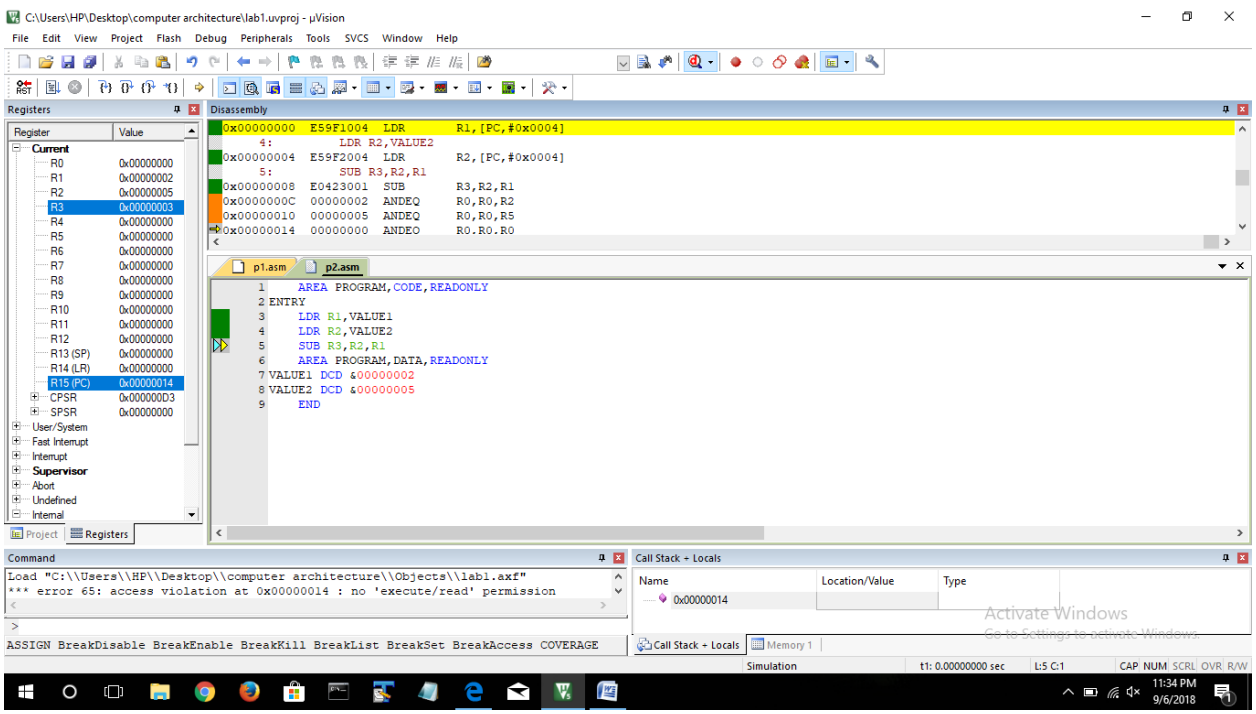
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,VALUE1
        LDR R2,VALUE2
        ADD R3,R2,R1
        AREA PROGRAM,DATA,READONLY
VALUE1 DCD &00000002
VALUE2 DCD &00000003
        END
```



SUBTRACTION OF TWO NUMBERS USING DIRECT ADDRESSING

CODE:

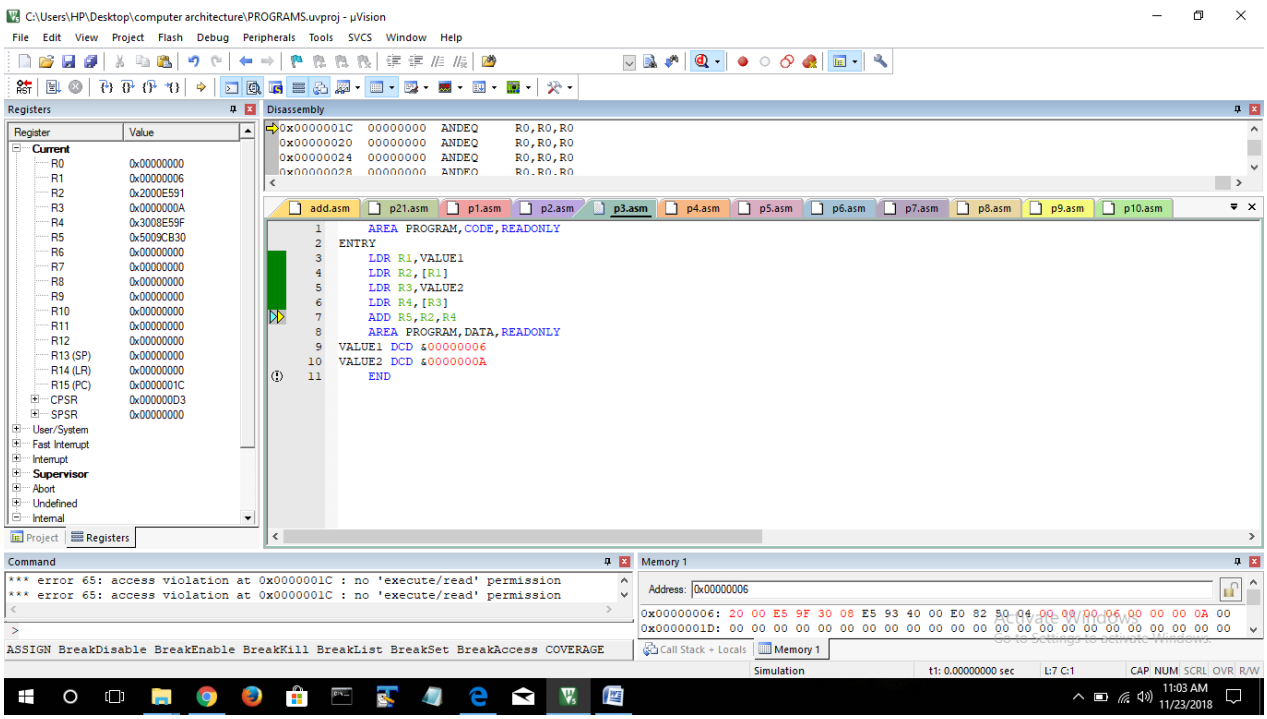
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,VALUE1
        LDR R2,VALUE2
        SUB R3,R2,R1
        AREA PROGRAM,DATA,READONLY
VALUE1 DCD &00000002
VALUE2 DCD &00000005
        END
```



ADDITION USING INDIRECT ADDRESSING

CODE:

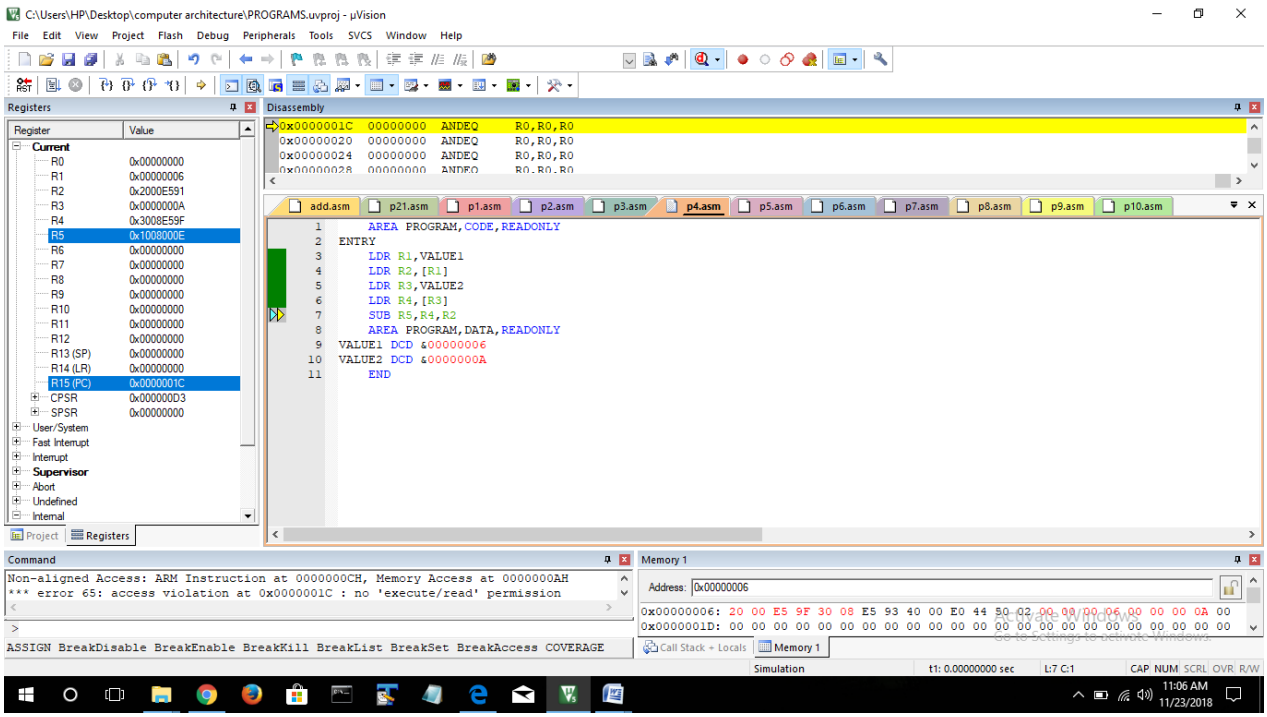
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,VALUE1
        LDR R2,[R1]
        LDR R3,VALUE2
        LDR R4,[R3]
        ADD R5,R2,R4
        AREA PROGRAM,DATA,READONLY
VALUE1 DCD &00000006
VALUE2 DCD &0000000A
        END
```



SUBTRACTION USING INDIRECT ADDRESSING

CODE

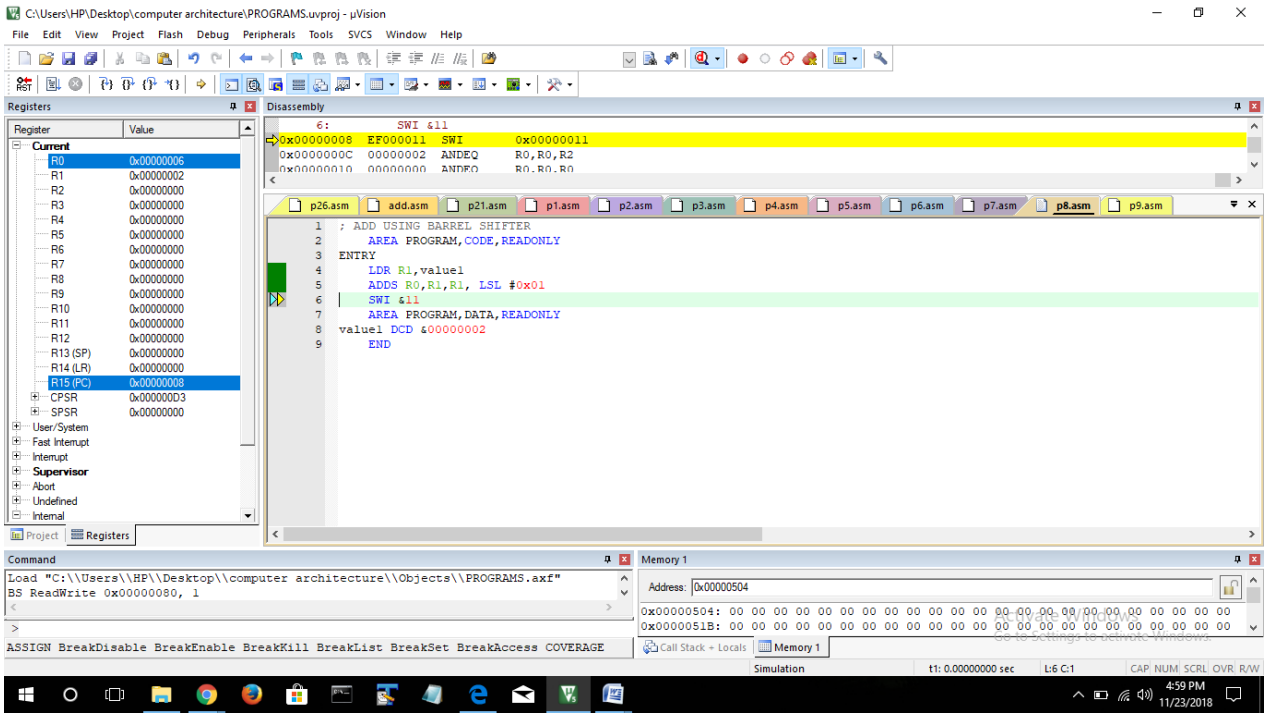
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,VALUE1
        LDR R2,[R1]
        LDR R3,VALUE2
        LDR R4,[R3]
        SUB R5,R4,R2
        AREA PROGRAM,DATA,READONLY
VALUE1 DCD &00000006
VALUE2 DCD &0000000A
        END
```



ADDITION USING BARREL SHIFTER

CODE:

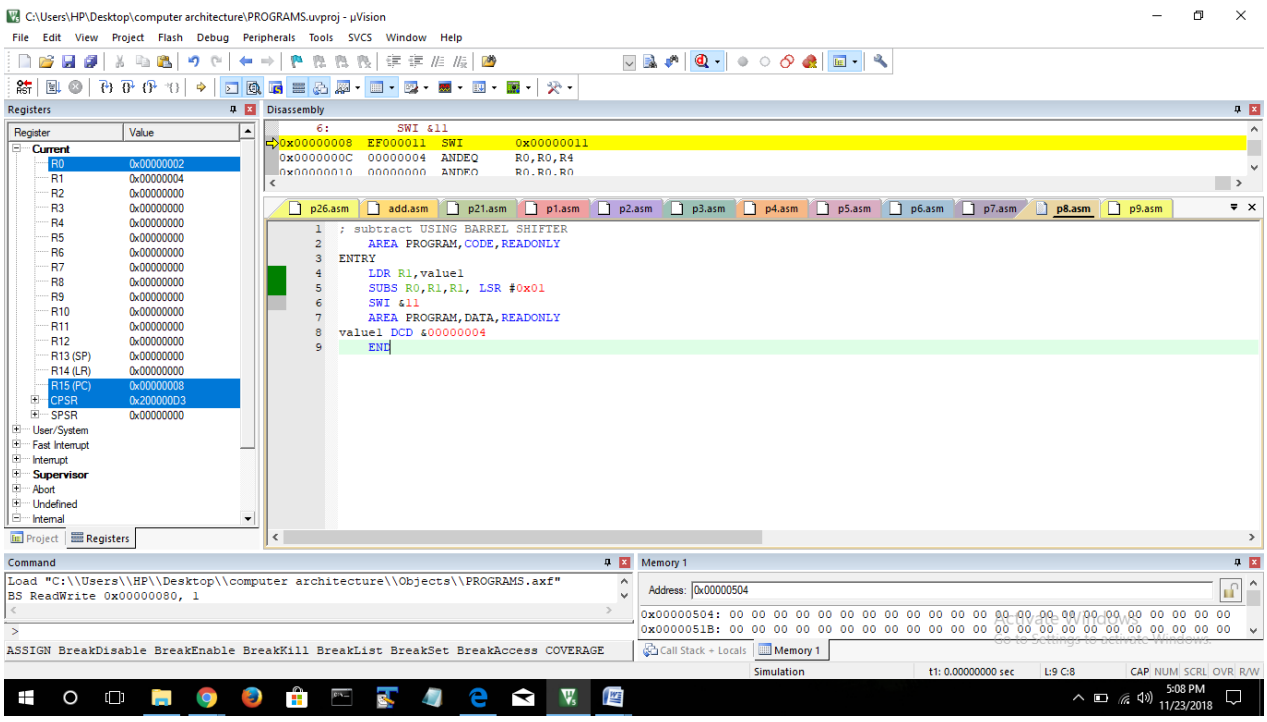
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,value1
        ADDS R0,R1,R1, LSL #0x01
        SWI &11
        AREA PROGRAM,DATA,READONLY
value1 DCD &00000002
        END
```



SUBTRACTION USING BARREL SHIFTER

CODE:

```
        AREA PROGRAM,CODE,READONLY
        ENTRY
                LDR R1,value1
                SUBS R0,R1,R1, LSR #0x01
                SWI &11
                AREA PROGRAM,DATA,READONLY
value1 DCD &00000004
                END
```

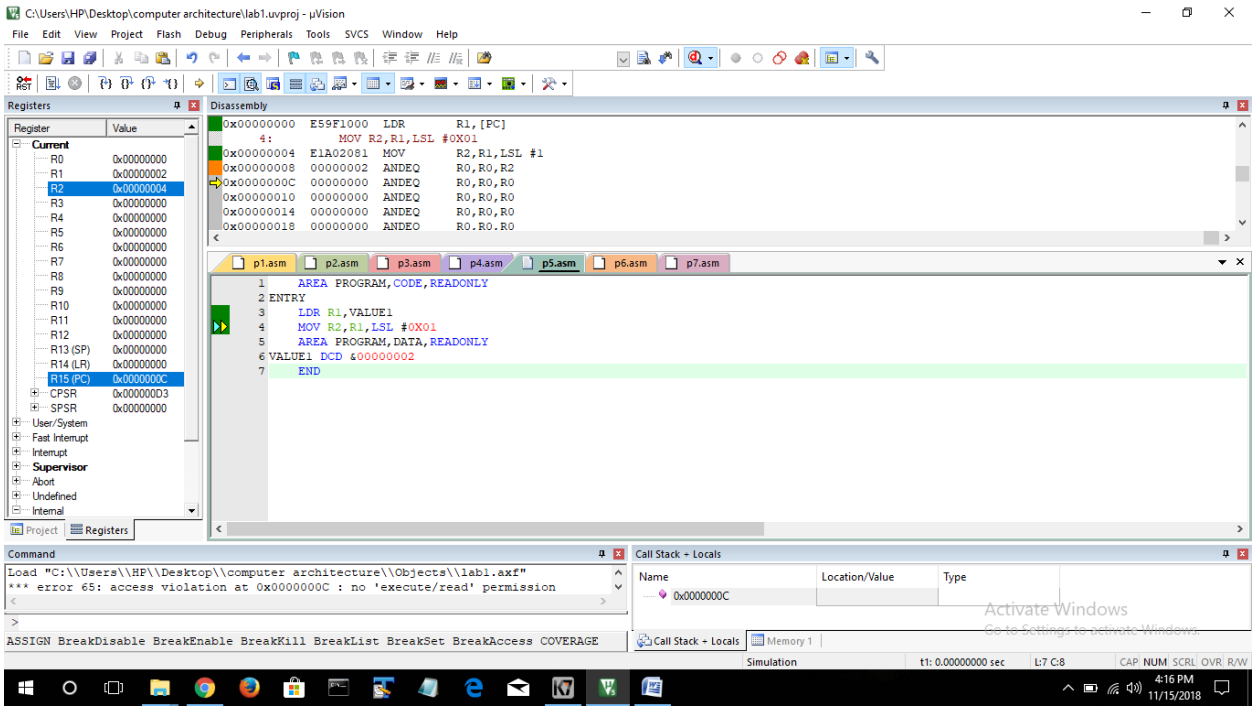


2. Write a program to perform left and right shift of a number.

LOGICAL SHIFT LEFT

CODE:

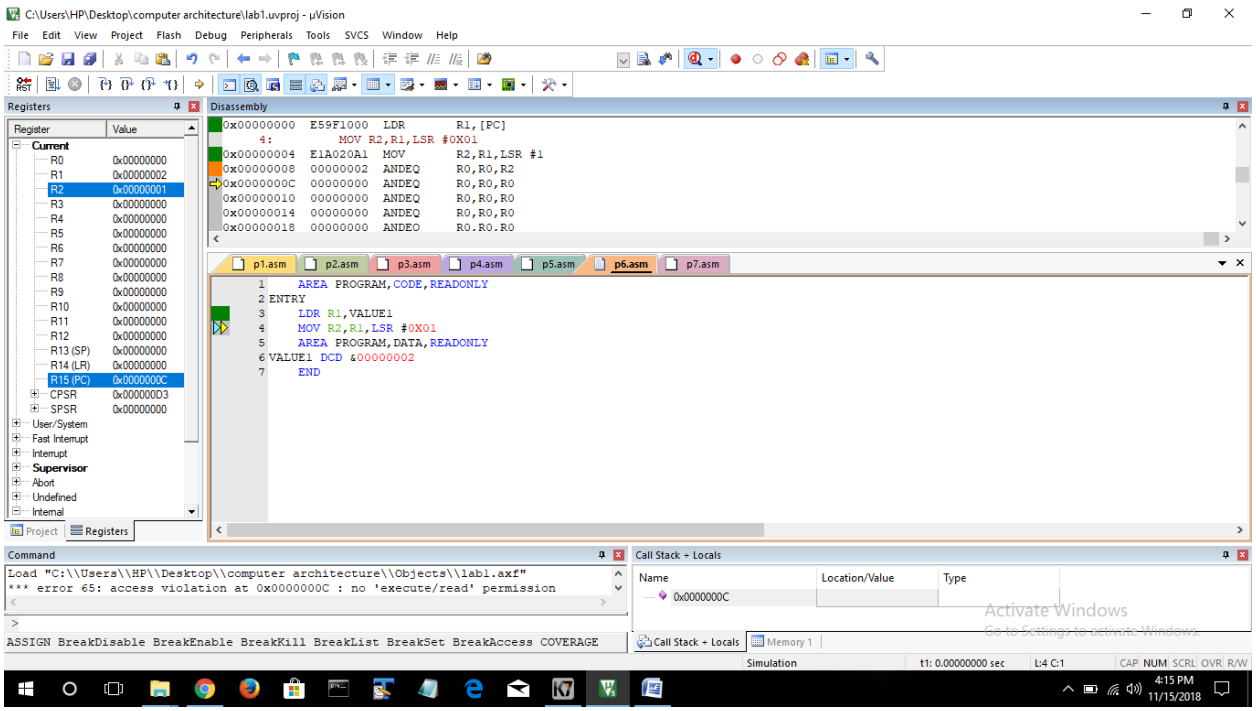
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,VALUE1
        MOV R2,R1,LSL #0X01
        AREA PROGRAM,DATA,READONLY
VALUE1 DCD &00000002
        END
```



LOGICAL SHIFT RIGHT

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,VALUE1
        MOV R2,R1,LSR #0X01
        AREA PROGRAM,DATA,READONLY
VALUE1 DCD &00000002
        END
```

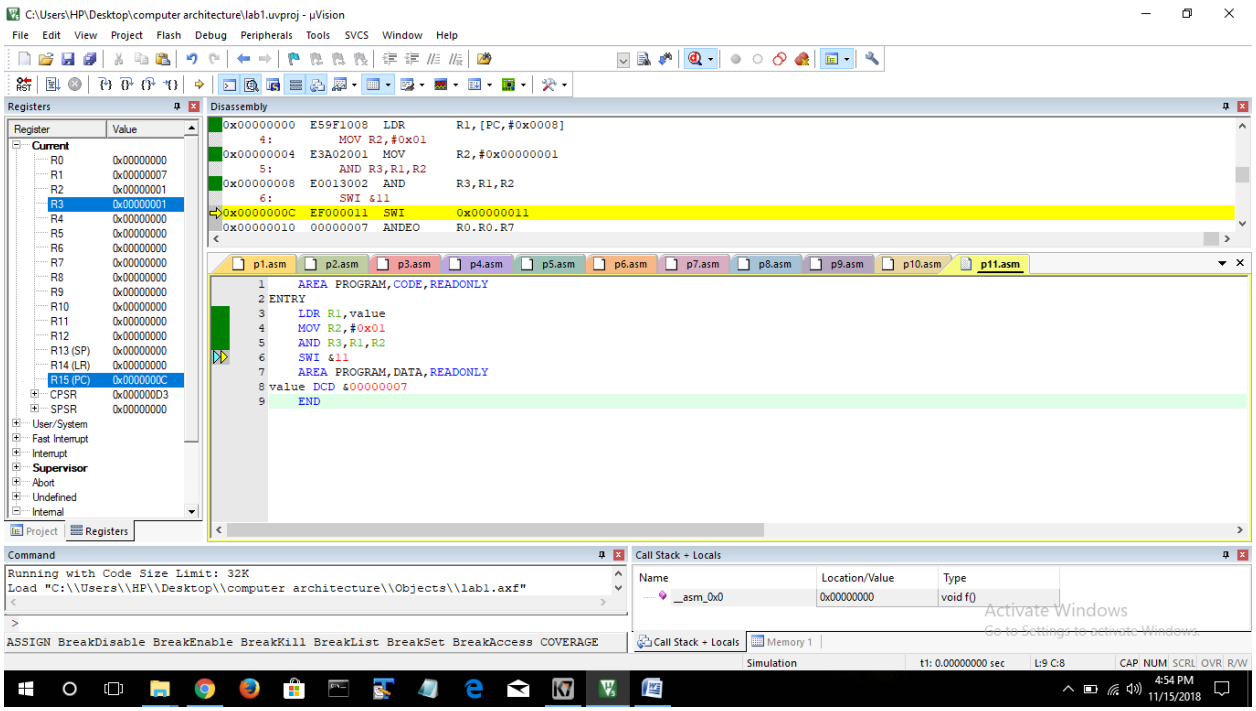


3. Write a program to find whether number is even or odd.

Number is even or odd

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,value
        MOV R2,#0x01
        AND R3,R1,R2
        SWI &11
        AREA PROGRAM,DATA,READONLY
value DCD &00000007
        END
```

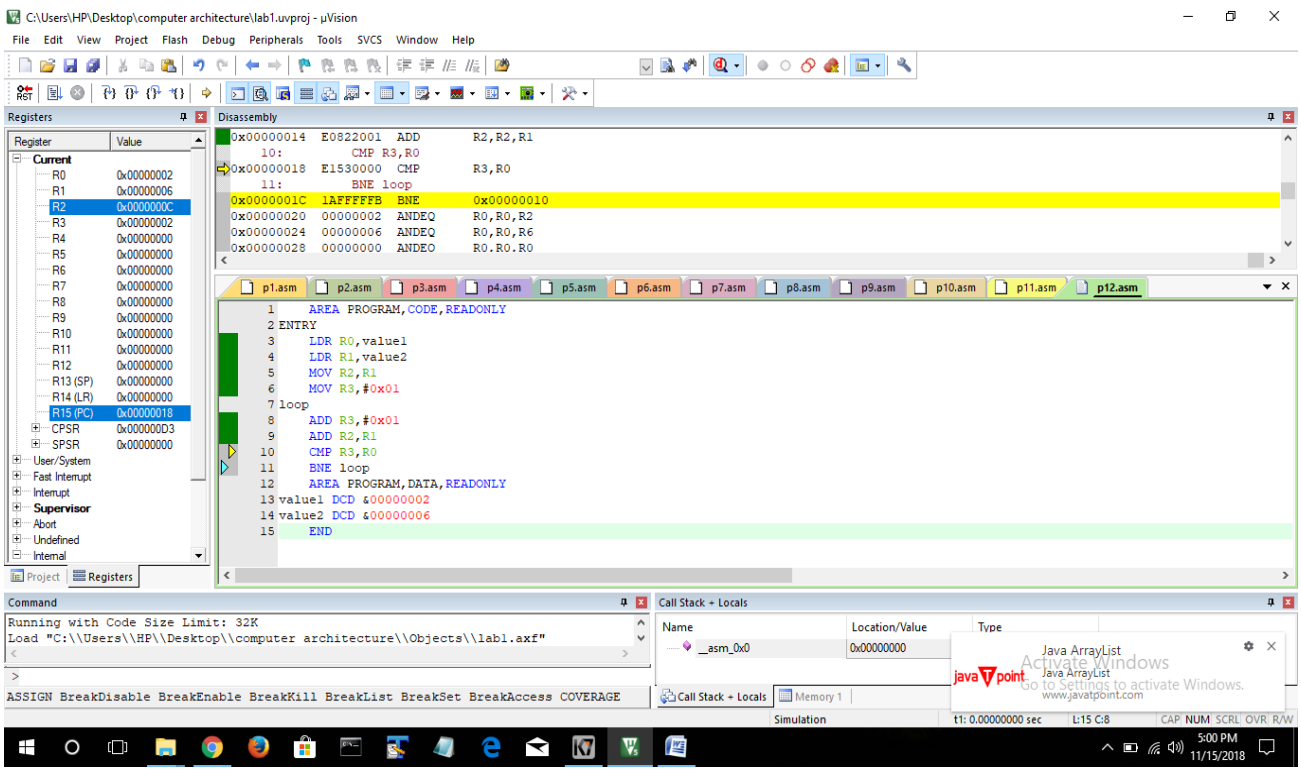


4. Write a program to perform Multiplication using addition.

Multiplication using addition

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,value1
        LDR R1,value2
        MOV R2,R1
        MOV R3,#0x01
loop
        ADD R3,#0x01
        ADD R2,R1
        CMP R3,R0
        BNE loop
        AREA PROGRAM,DATA,READONLY
value1 DCD &00000002
value2 DCD &00000006
        END
```

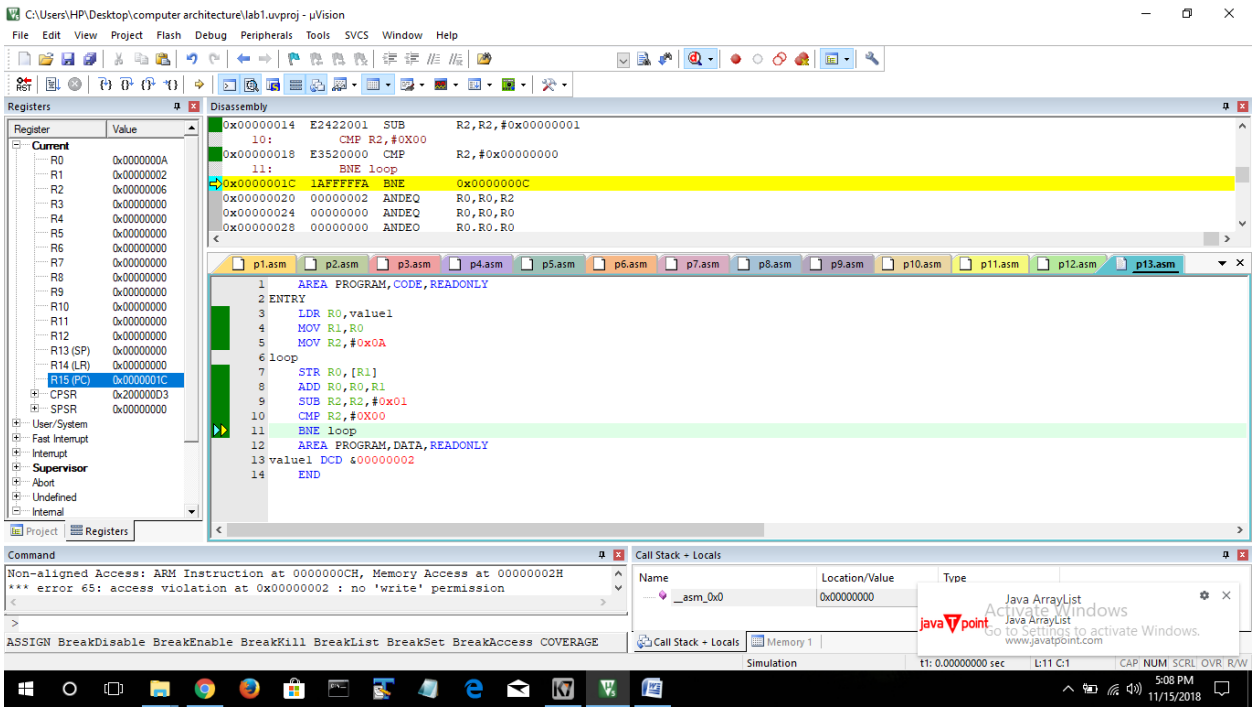


5. Write a program to store Multiplication table of a number.

Table of a no.

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,value1
        MOV R1,R0
        MOV R2,#0x0A
loop
        STR R0,[R1]
        ADD R0,R0,R1
        SUB R2,R2,#0x01
        CMP R2,#0X00
        BNE loop
        AREA PROGRAM,DATA,READONLY
value1 DCD &00000002
        END
```

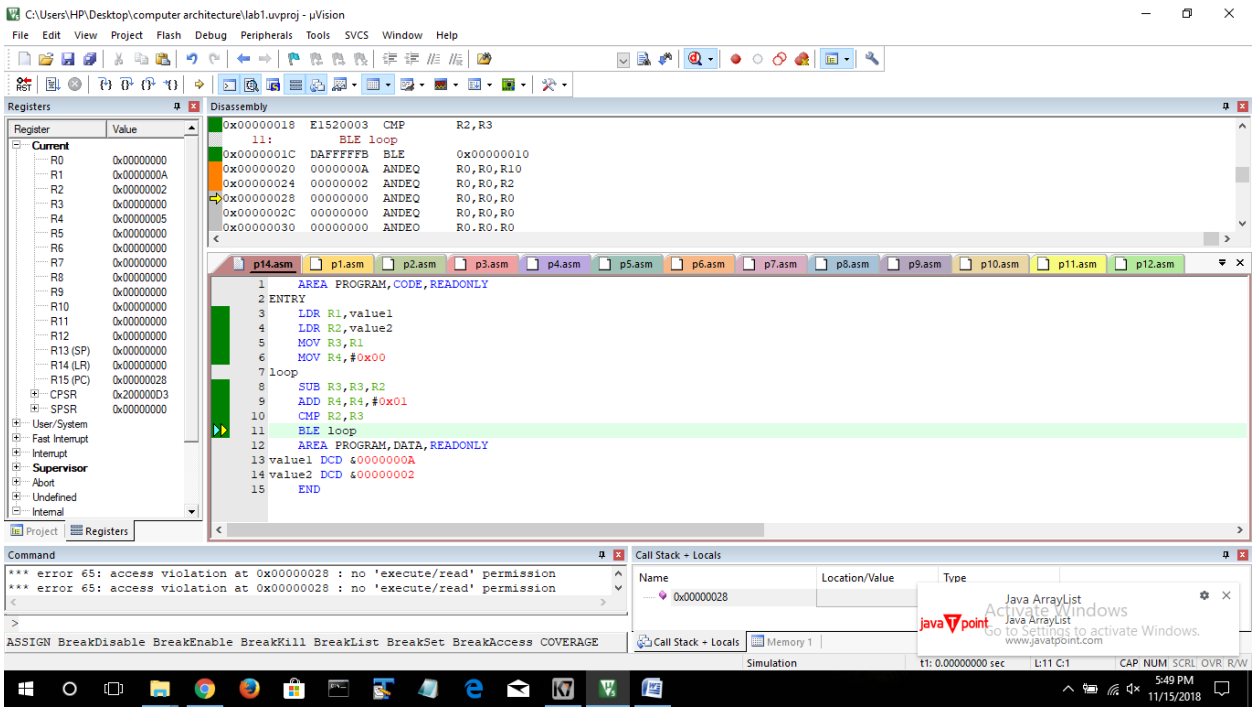


6. Write a program to perform Division using subtraction.

Division using subtraction

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,value1
        LDR R2,value2
        MOV R3,R1
        MOV R4,#0x00
loop
        SUB R3,R3,R2
        ADD R4,R4,#0x01
        CMP R2,R3
        BLE loop
        AREA PROGRAM,DATA,READONLY
value1 DCD &0000000A
value2 DCD &00000002
        END
```

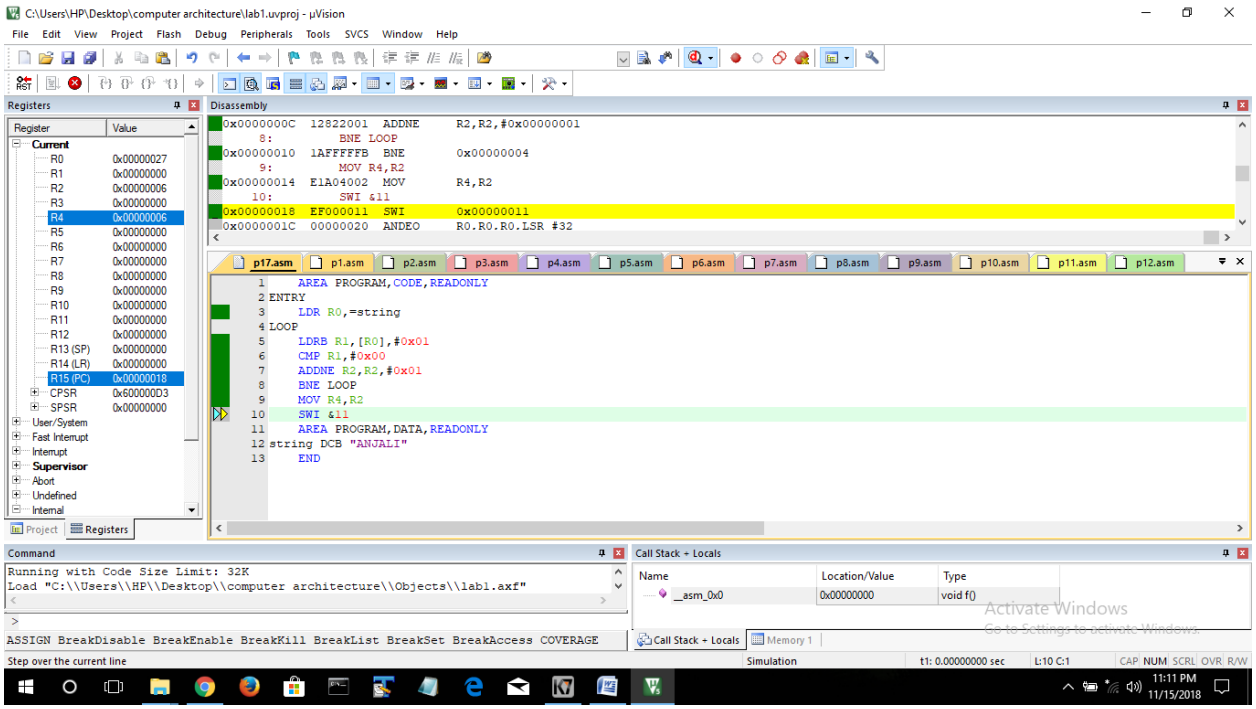


7. Write a program to count the number of characters in a given string.

Count no. of characters in string

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,=string
LOOP
        LDRB R1,[R0],#0x01
        CMP R1,#0x00
        ADDNE R2,R2,#0x01
        BNE LOOP
        MOV R4,R2
        SWI &11
        AREA PROGRAM,DATA,READONLY
string DCB "ANJALI"
        END
```

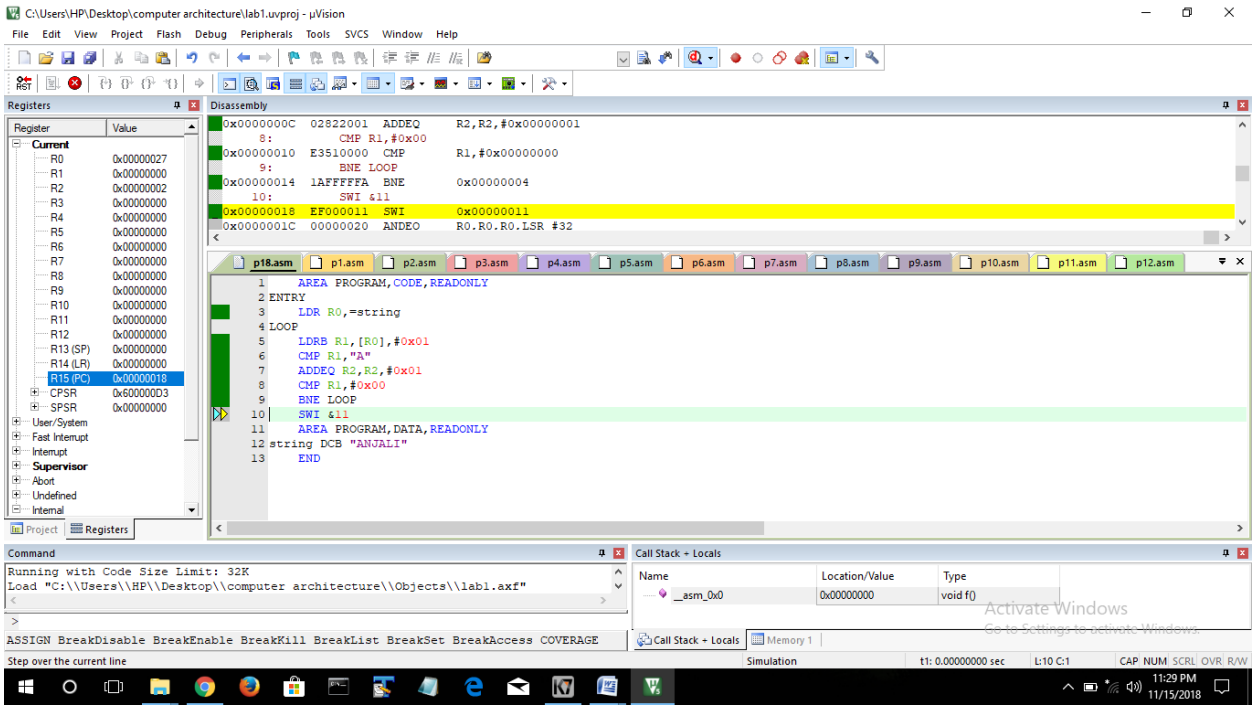


8. Write a program to find the number of occurrence of a particular character in a string.

COUNT A PARTICULAR CHARACTER IN A STRING

CODE:

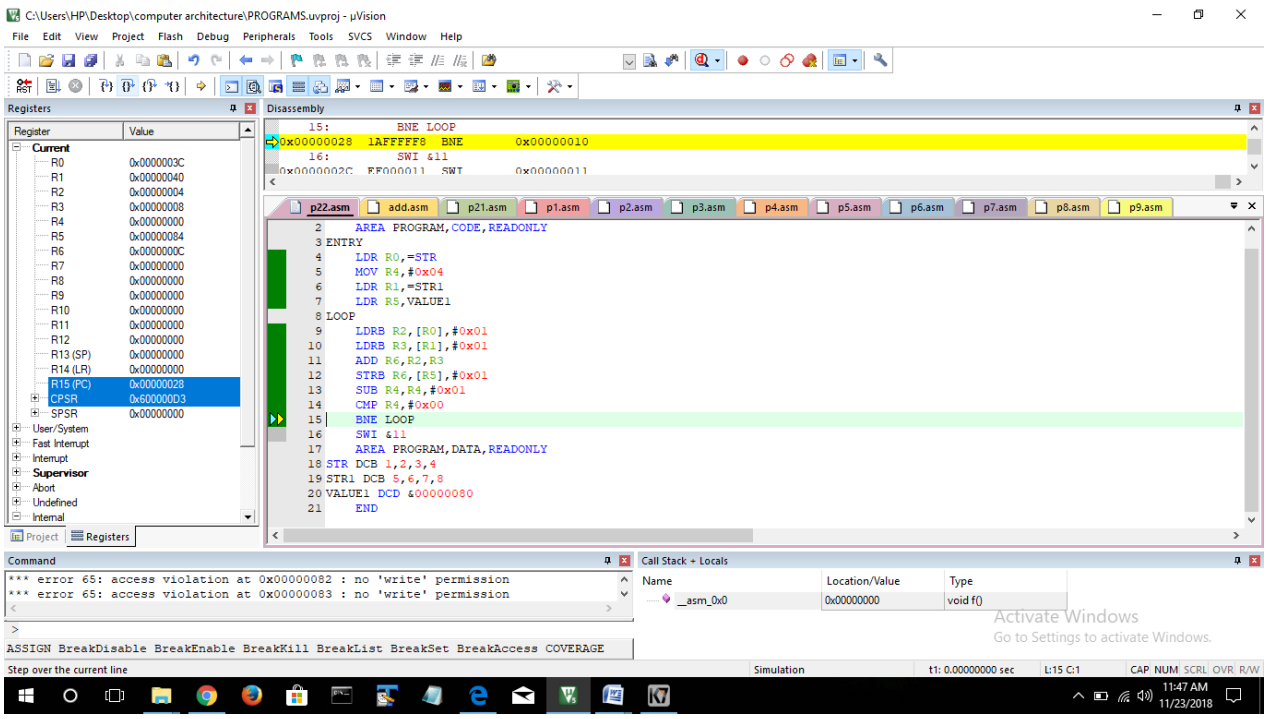
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,=string
LOOP
        LDRB R1,[R0],#0x01
        CMP R1,"A"
        ADDEQ R2,R2,#0x01
        CMP R1,#0x00
        BNE LOOP
        SWI &11
        AREA PROGRAM,DATA,READONLY
string DCB "ANJALI"
        END
```



9. Write a program to add two integer strings.

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,=STR
        MOV R4,#0x04
        LDR R1,=STR1
        LDR R5,VALUE1
LOOP
        LDRB R2,[R0],#0x01
        LDRB R3,[R1],#0x01
        ADD R6,R2,R3
        STRB R6,[R5],#0x01
        SUB R4,R4,#0x01
        CMP R4,#0x00
        BNE LOOP
        SWI &11
        AREA PROGRAM,DATA,READONLY
STR DCB 1,2,3,4
STR1 DCB 5,6,7,8
VALUE1 DCD &00000080
        END
```

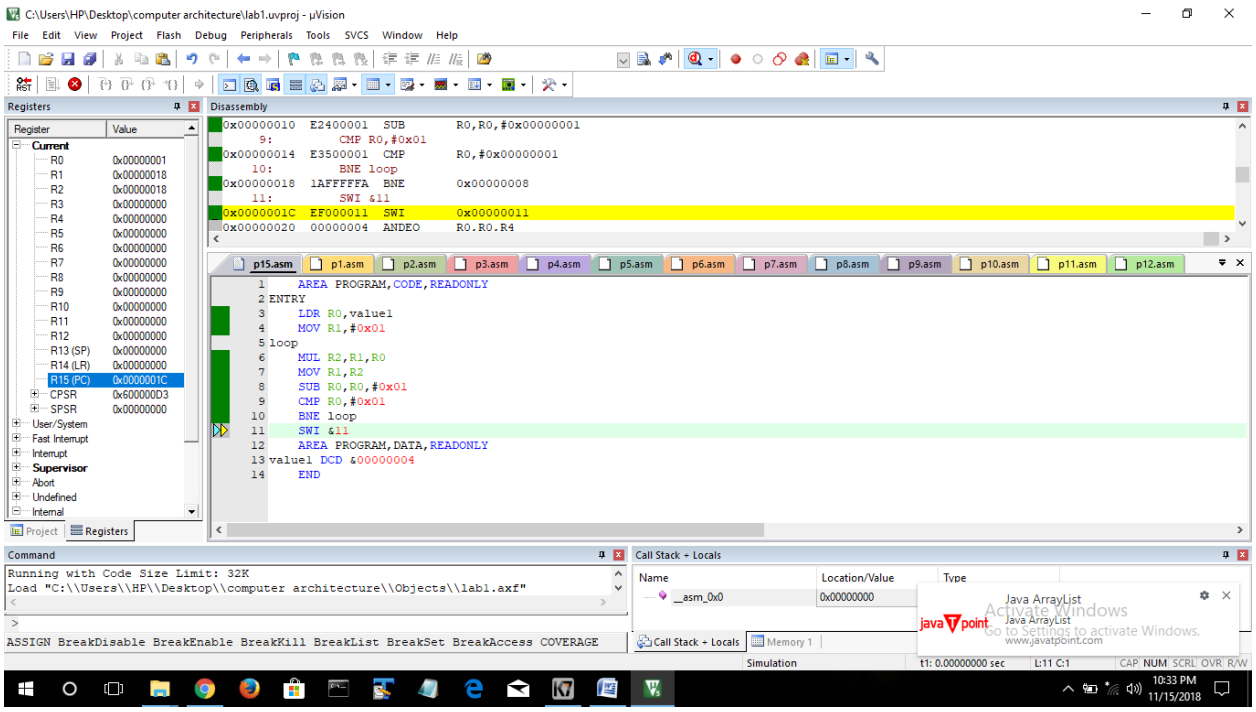


10. Write a program to find the factorial of a number.

Factorial of no.

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,value1
        MOV R1,#0x01
loop
        MUL R2,R1,R0
        MOV R1,R2
        SUB R0,R0,#0x01
        CMP R0,#0x01
        BNE loop
        SWI &11
        AREA PROGRAM,DATA,READONLY
value1 DCD &00000004
        END
```

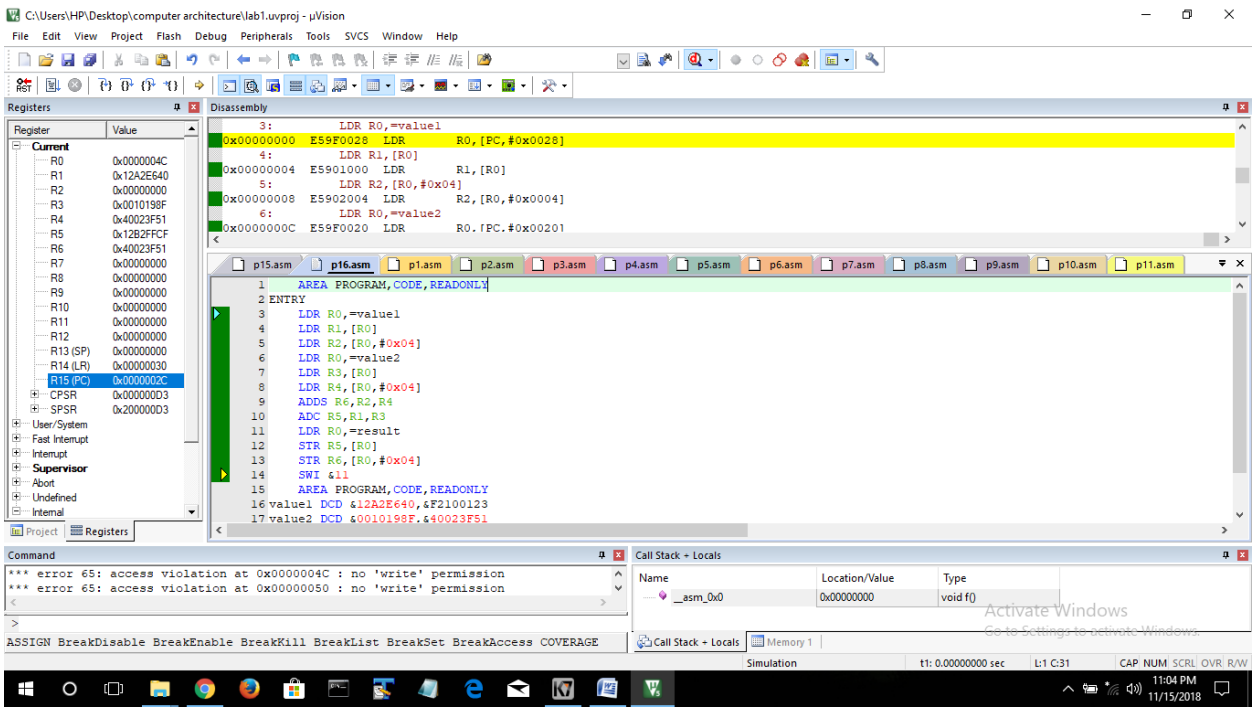


11. Write a program to perform addition of two 64-bit numbers.

64- bit Addition

CODE:

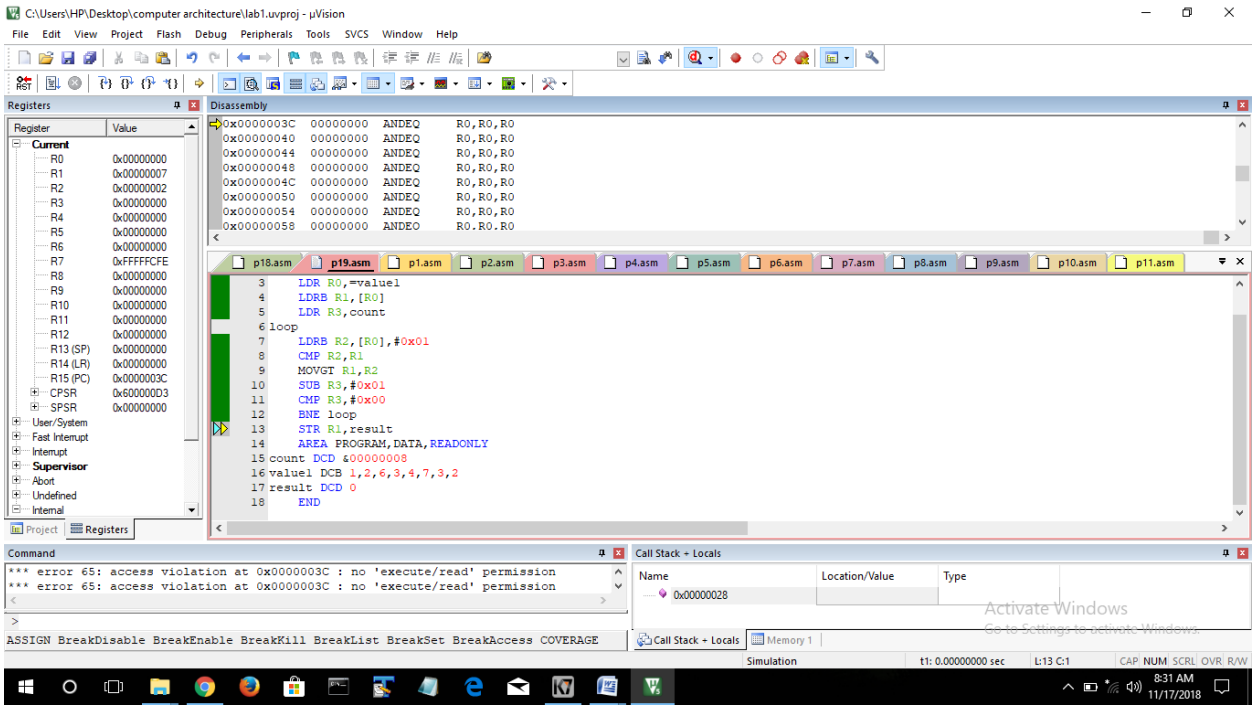
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,=value1
        LDR R1,[R0]
        LDR R2,[R0,#0x04]
        LDR R0,=value2
        LDR R3,[R0]
        LDR R4,[R0,#0x04]
        ADDS R6,R2,R4
        ADC R5,R1,R3
        LDR R0,=result
        STR R5,[R0]
        STR R6,[R0,#0x04]
        SWI &11
        AREA PROGRAM,CODE,READONLY
value1 DCD &12A2E640,&F2100123
value2 DCD &0010198F,&40023F51
result DCD 0
        END
```



12. Write a program to find the largest number in an array.

CODE:

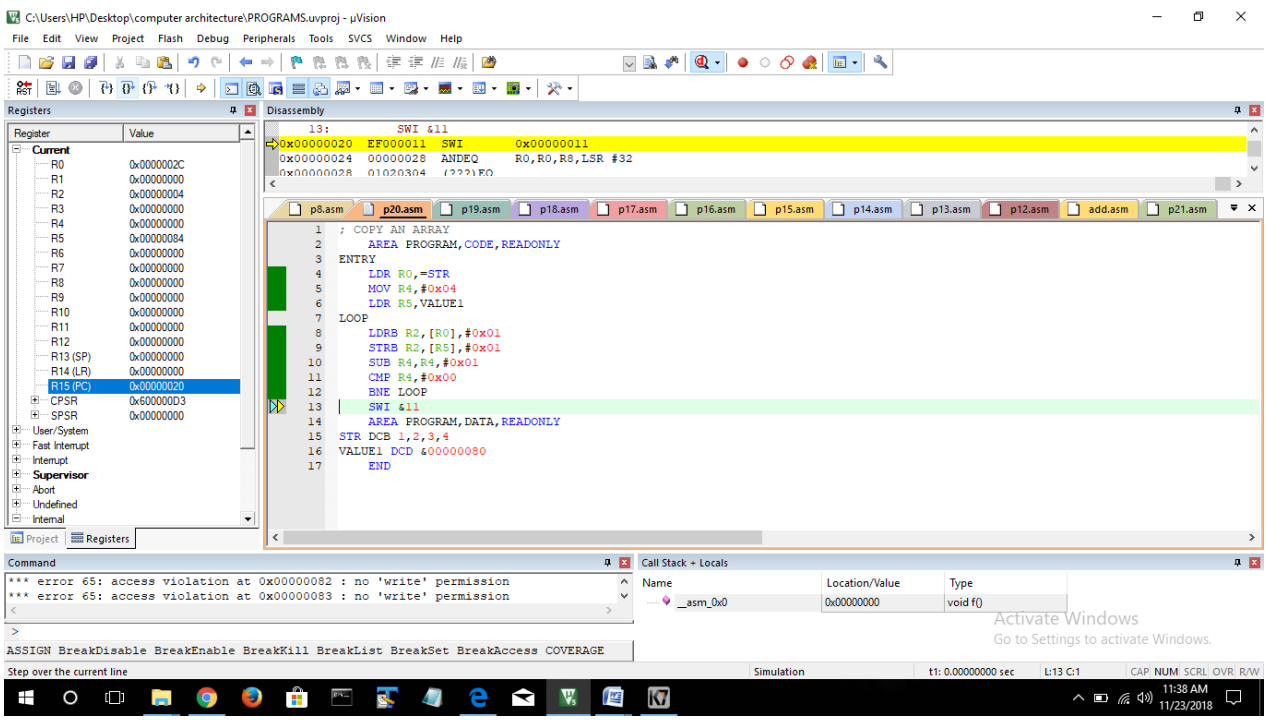
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,=value1
        LDRB R1,[R0]
        LDR R3,count
loop
        LDRB R2,[R0],#0x01
        CMP R2,R1
        MOVGT R1,R2
        SUB R3,#0x01
        CMP R3,#0x00
        BNE loop
        STR R1,result
        AREA PROGRAM,DATA,READONLY
count DCD &00000008
value1 DCB 1,2,6,3,4,7,3,2
result DCD 0
        END
```



13. Write a program to copy an array.

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,=STR
        MOV R4,#0x04
        LDR R5,VALUE1
LOOP
        LDRB R2,[R0],#0x01
        STRB R2,[R5],#0x01
        SUB R4,R4,#0x01
        CMP R4,#0x00
        BNE LOOP
        SWI &11
        AREA PROGRAM,DATA,READONLY
STR DCB 1,2,3,4
VALUE1 DCD &00000080
        END
```

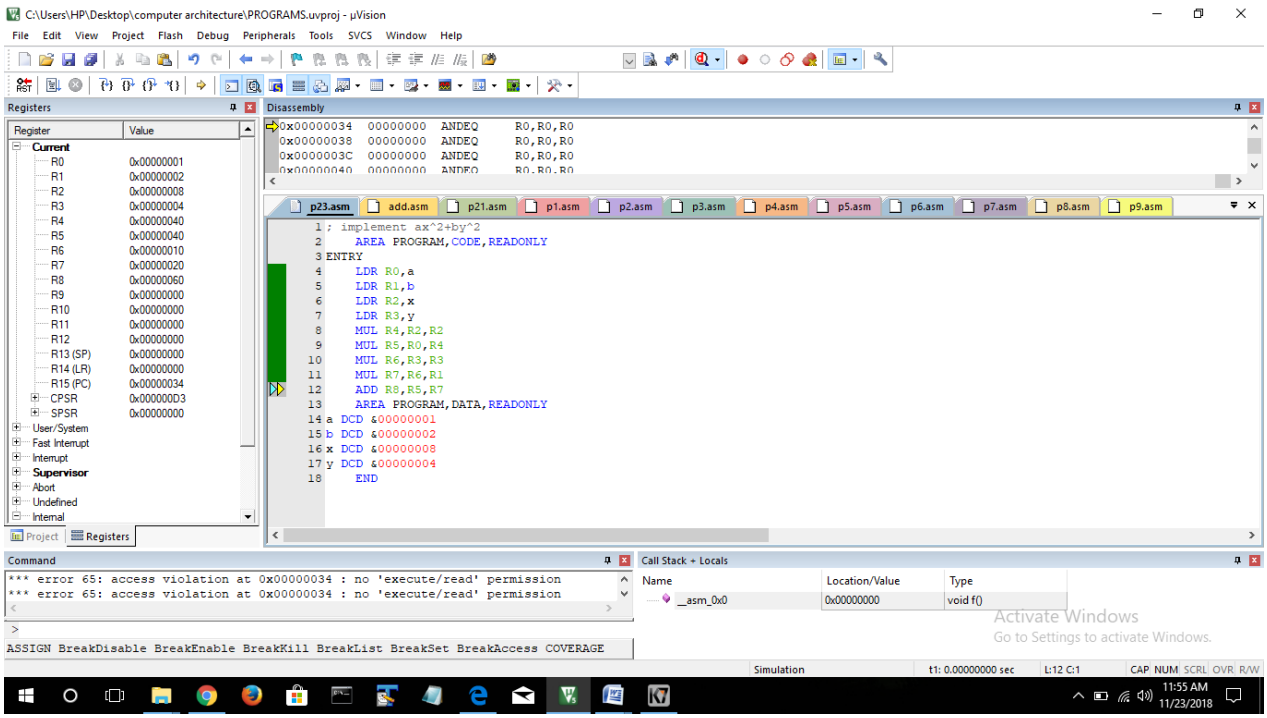


14. Write a program in ARM assembly language to implement the following equation:

i) ax2+by2

CODE:

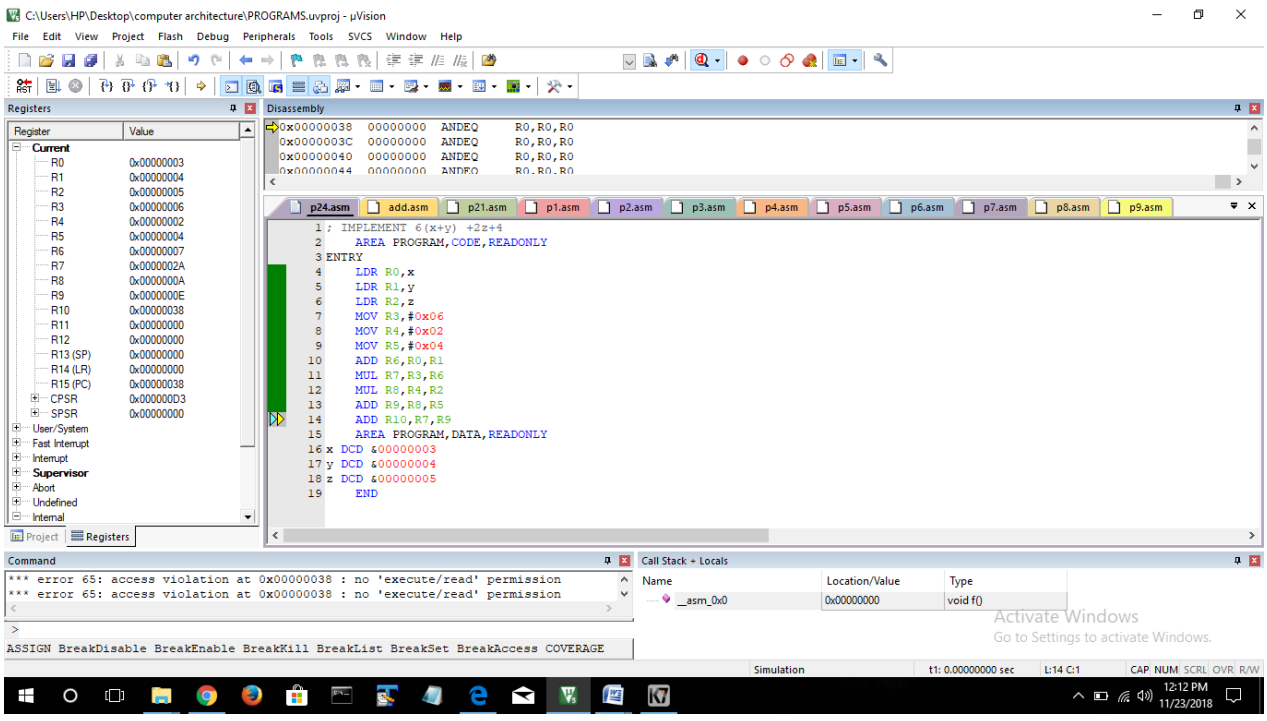
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,a
        LDR R1,b
        LDR R2,x
        LDR R3,y
        MUL R4,R2,R2
        MUL R5,R0,R4
        MUL R6,R3,R3
        MUL R7,R6,R1
        ADD R8,R5,R7
        AREA PROGRAM,DATA,READONLY
a DCD &00000001
b DCD &00000002
x DCD &00000008
y DCD &00000004
        END
```



ii) 6(x+y) +2z+4

CODE:

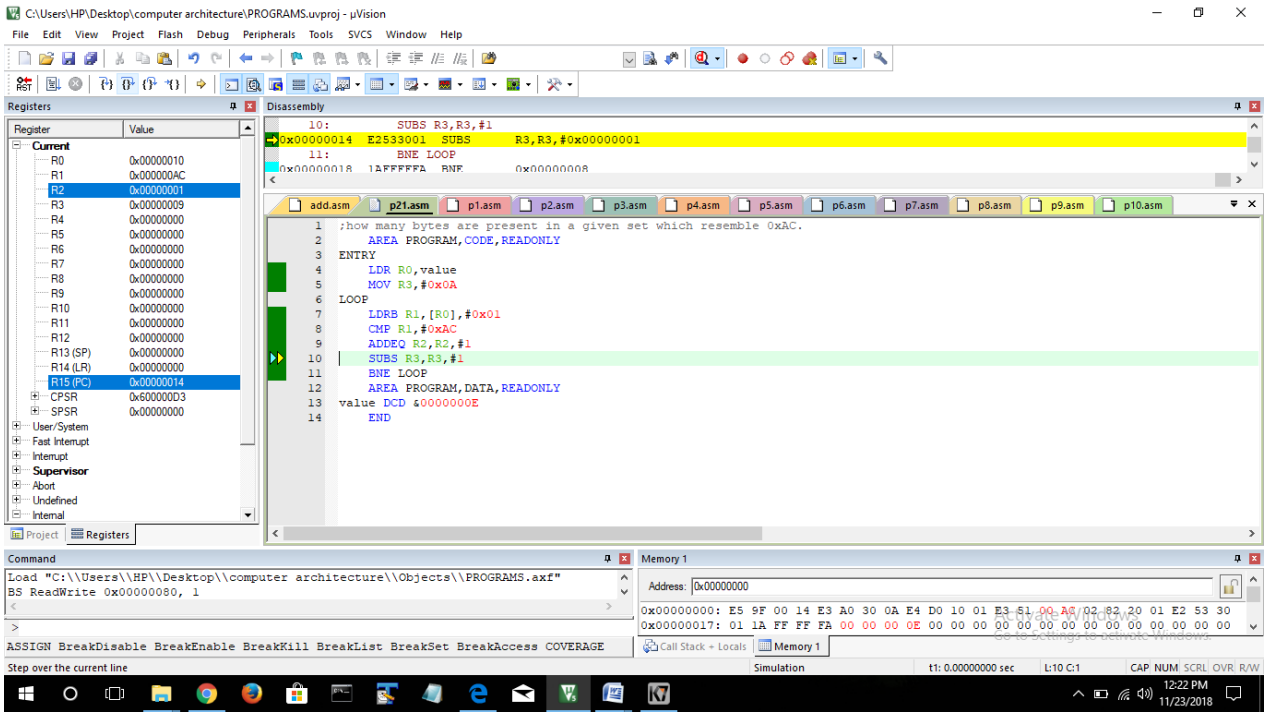
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,x
        LDR R1,y
        LDR R2,z
        MOV R3,#0x06
        MOV R4,#0x02
        MOV R5,#0x04
        ADD R6,R0,R1
        MUL R7,R3,R6
        MUL R8,R4,R2
        ADD R9,R8,R5
        ADD R10,R7,R9
        AREA PROGRAM,DATA,READONLY
x DCD &00000003
y DCD &00000004
z DCD &00000005
        END
```



15. Write a program in ARM assembly language to verify how many bytes are present in a given set which resemble 0xAC.

CODE:

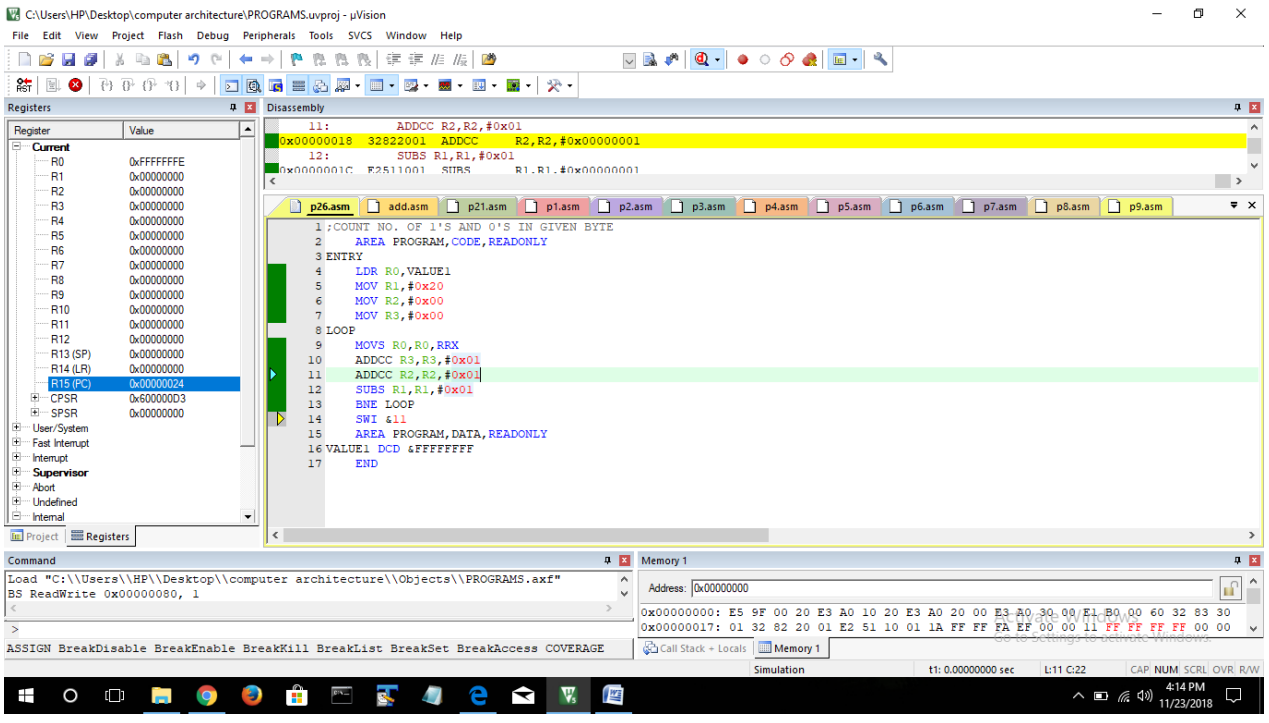
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,value
        MOV R3,#0x0A
LOOP
        LDRB R1,[R0],#0x01
        CMP R1,#0xAC
        ADDEQ R2,R2,#1
        SUBS R3,R3,#1
        BNE LOOP
        AREA PROGRAM,DATA,READONLY
value DCD &0000000E
        END
```



16. Write a program in ARM assembly language to count the number of 1s and 0s in a given byte and verify the result.

CODE:

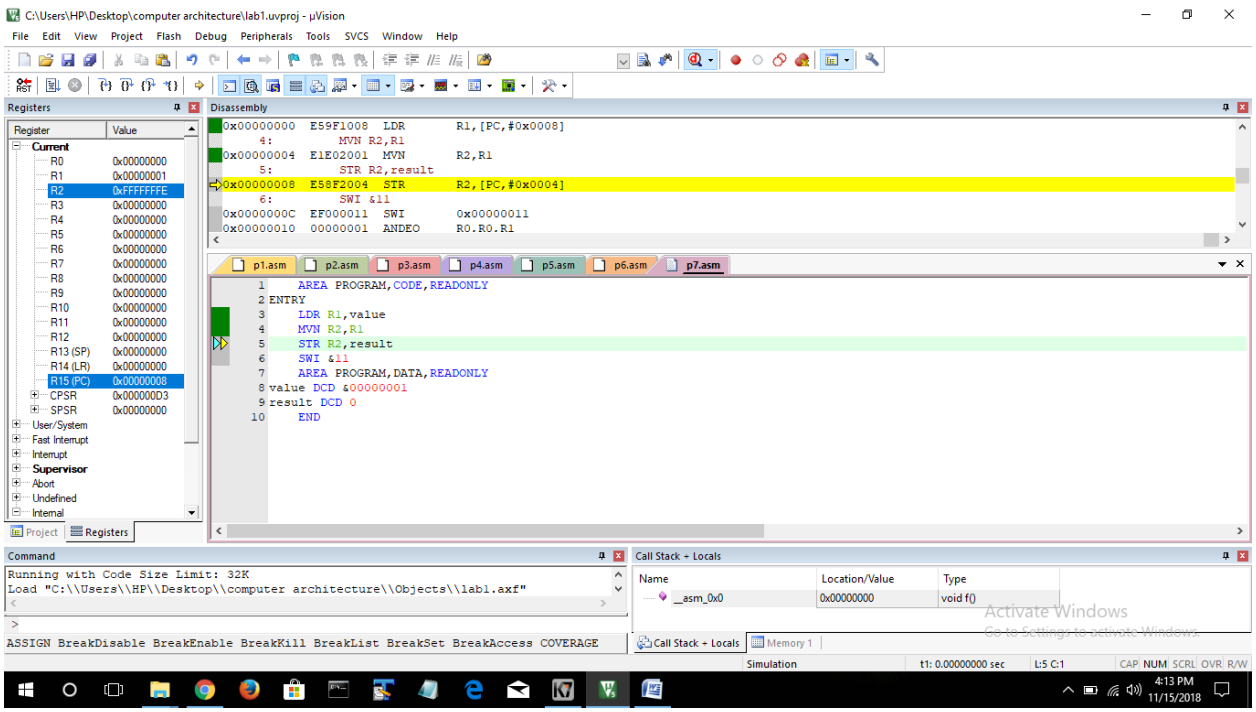
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R0,VALUE1
        MOV R1,#0x20
        MOV R2,#0x00
        MOV R3,#0x00
LOOP
        MOVS R0,R0,RRX
        ADDCS R3,R3,#0x01
        ADDCC R2,R2,#0x01
        SUBS R1,R1,#0x01
        BNE LOOP
        SWI &11
        AREA PROGRAM,DATA,READONLY
VALUE1 DCD &FFFFFFFF
        END
```



One’s Comlement

CODE:

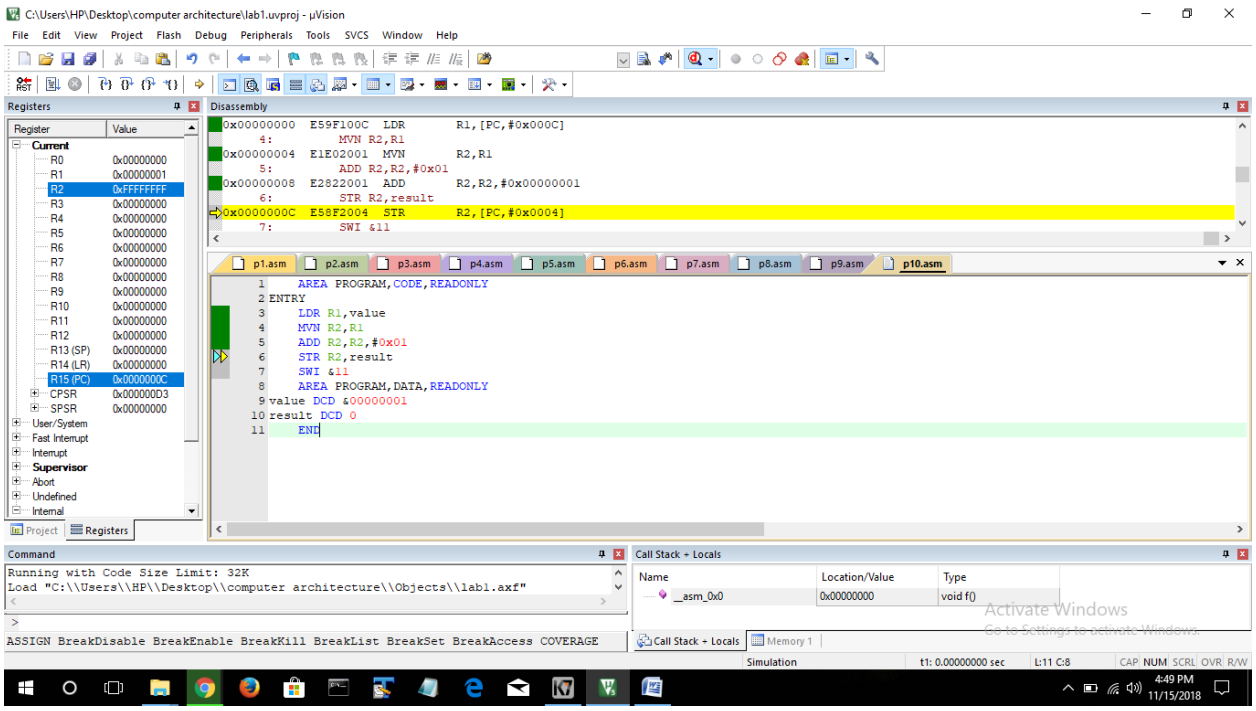
```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,value
        MVN R2,R1
        STR R2,result
        SWI &11
        AREA PROGRAM,DATA,READONLY
value DCD &00000001
result DCD 0
        END
```



2’s complement

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,value
        MVN R2,R1
        ADD R2,R2,#0x01
        STR R2,result
        SWI &11
        AREA PROGRAM,DATA,READONLY
value DCD &00000001
result DCD 0
        END
```



Greater of two no.

CODE:

```
        AREA PROGRAM,CODE,READONLY
ENTRY
        LDR R1,value1
        LDR R2,value2
        CMP R1,R2
        BHI abcd
        MOV R1,R2
abcd
        STR R1,result
        SWI &11
        AREA PROGRAM,DATA,READONLY
value1 DCD &00000003
value2 DCD &00000006
result DCD 0
        END
```