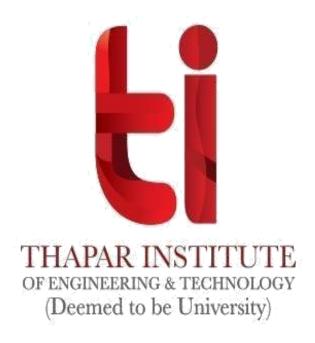
MICROPROCESSOR BASED SYSTEMS DESIGN

UCS617

LAB ASSIGNMENT ARM MICROPROCESSOR



Submitted By:

Rishabh Kumar 101611046 Rohan Subedi 101603284

Submitted To:

Ms. Swadha Gupta

INDEX

S.No.	Name of Practical's	Page No.
1	Write a program in ARM assembly language to add and subtract two	3-8
	32-bit numbers using: i) Direct addressing mode ii) Indirect addressing	
	mode iii) Barrel shifter	
2	Write a program to perform left and right shift of a number.	9-10
3	Write a program to find whether number is even or odd.	11
4	Write a program to perform Multiplication using addition.	12
5	Write a program to store Multiplication table of a number.	13
6	Write a program to perform Division using subtraction.	14
7	Write a program to count the number of characters in a given string.	15
8	Write a program to find the number of occurrence of a particular	16
	character in a string.	
9	Write a program to add two integer strings.	17
10	Write a program to find the factorial of a number.	18
11	Write a program to perform addition of two 64-bit numbers.	19
12	Write a program to find the largest number in an array.	20
13	Write a program to copy an array.	21
14	Write a program in ARM assembly language to implement the	22-23
	following equation: i) $ax2+by2$ ii) $6(x+y)+2z+4$	
15	Write a program in ARM assembly language to verify how many bytes	24
	are present in a given set which resemble 0xAC.	
16	Write a program in ARM assembly language to count the number of 1s	25
	and 0s in a given byte and verify the result.	
17	Write a program in ARM assembly language to find One's complement	26
	of a number.	
18	Write a program in ARM assembly language to find Two's	27
	complement of a number.	
19	Write a program in ARM assembly language to find greater of two	28
	numbers.	

- 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

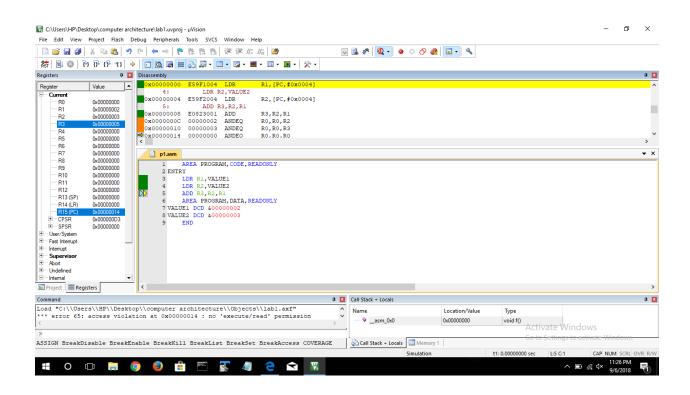
LDR R1,VALUE1 LDR R2,VALUE2

ADD R3,R2,R1

AREA PROGRAM, DATA, READONLY

VALUE1 DCD &00000002

VALUE2 DCD &00000003



SUBTRACTION OF TWO NUMBERS USING DIRECT ADDRESSING

CODE:

AREA PROGRAM, CODE, READONLY

ENTRY

LDR R1, VALUE1

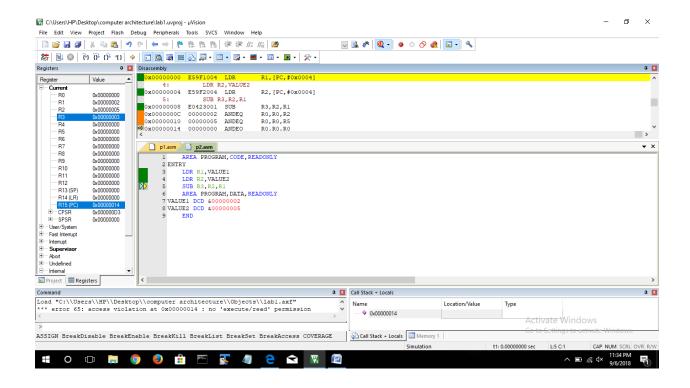
LDR R2, VALUE2

SUB R3,R2,R1

AREA PROGRAM, DATA, READONLY

VALUE1 DCD &00000002

VALUE2 DCD &00000005



ADDITION USING INDIRECT ADDRESSING

CODE:

AREA PROGRAM, CODE, READONLY

ENTRY

LDR R1, VALUE1

LDR R2,[R1]

LDR R3, VALUE2

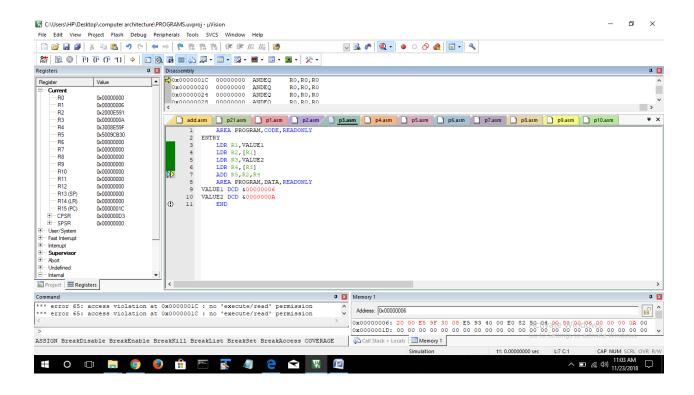
LDR R4,[R3]

ADD R5,R2,R4

AREA PROGRAM, DATA, READONLY

VALUE1 DCD &00000006

VALUE2 DCD &0000000A



SUBTRACTION USING INDIRECT ADDRESSING

CODE

AREA PROGRAM, CODE, READONLY

ENTRY

LDR R1, VALUE1

LDR R2,[R1]

LDR R3, VALUE2

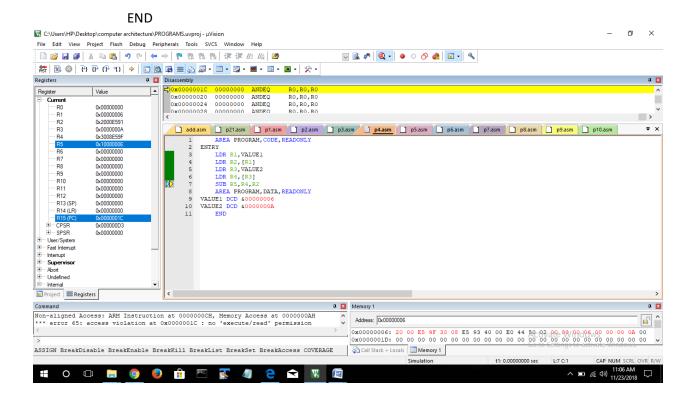
LDR R4,[R3]

SUB R5,R4,R2

AREA PROGRAM, DATA, READONLY

VALUE1 DCD &00000006

VALUE2 DCD &0000000A



ADDITION USING BARREL SHIFTER

CODE:

AREA PROGRAM, CODE, READONLY

ENTRY

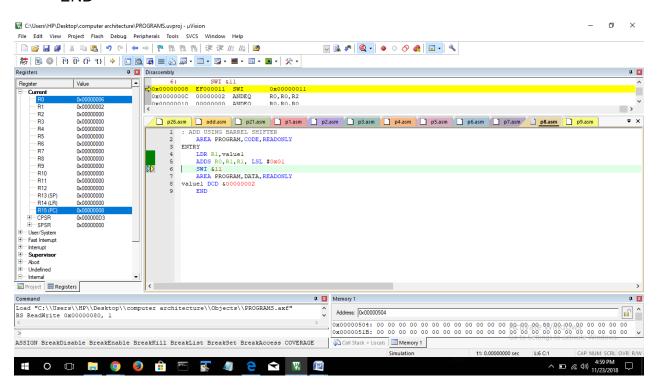
LDR R1,value1

ADDS R0,R1,R1, LSL #0x01

SWI &11

AREA PROGRAM, DATA, READONLY

value1 DCD &00000002



SUBTRACTION USING BARREL SHIFTER

CODE:

AREA PROGRAM, CODE, READONLY

ENTRY

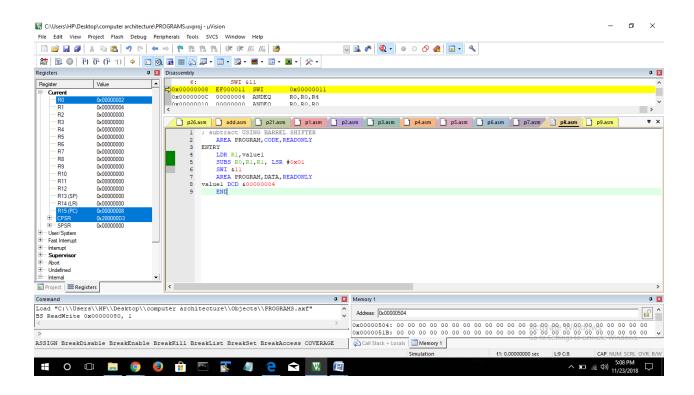
LDR R1, value1

SUBS R0,R1,R1, LSR #0x01

SWI &11

AREA PROGRAM, DATA, READONLY

value1 DCD &00000004



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

LOGICAL SHIFT LEFT

CODE:

AREA PROGRAM, CODE, READONLY

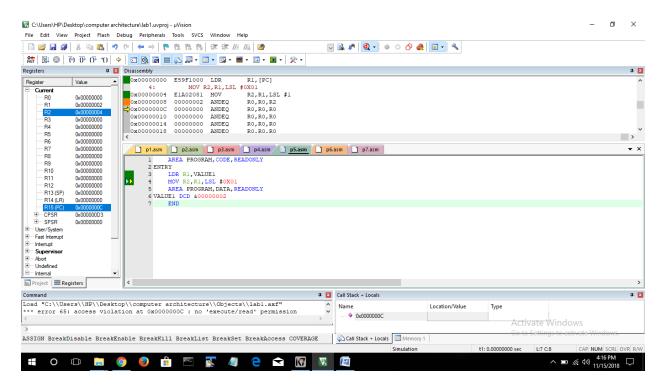
ENTRY

LDR R1, VALUE1

MOV R2,R1,LSL #0X01

AREA PROGRAM, DATA, READONLY

VALUE1 DCD &00000002



LOGICAL SHIFT RIGHT

CODE:

AREA PROGRAM, CODE, READONLY

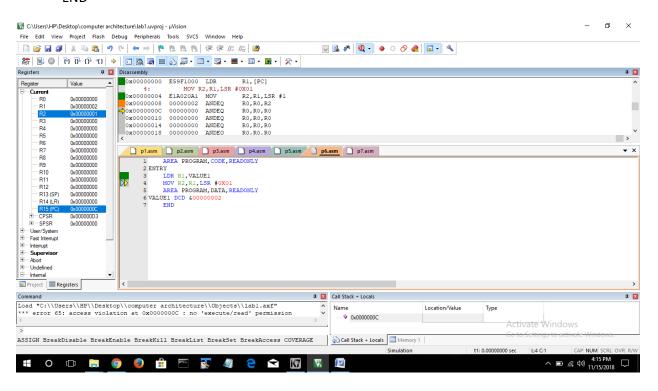
ENTRY

LDR R1, VALUE1

MOV R2,R1,LSR #0X01

AREA PROGRAM, DATA, READONLY

VALUE1 DCD &00000002



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

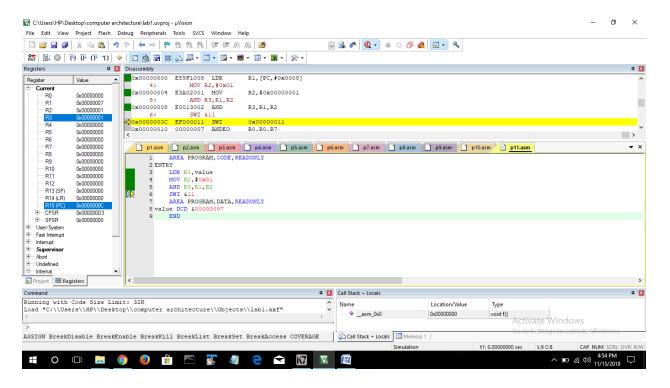
MOV R2,#0x01

AND R3,R1,R2

SWI &11

AREA PROGRAM, DATA, READONLY

value DCD &00000007



4. Write a program to perform Multiplication using addition.

Multiplication using addition

CODE:

```
AREA PROGRAM,CODE,READONLY

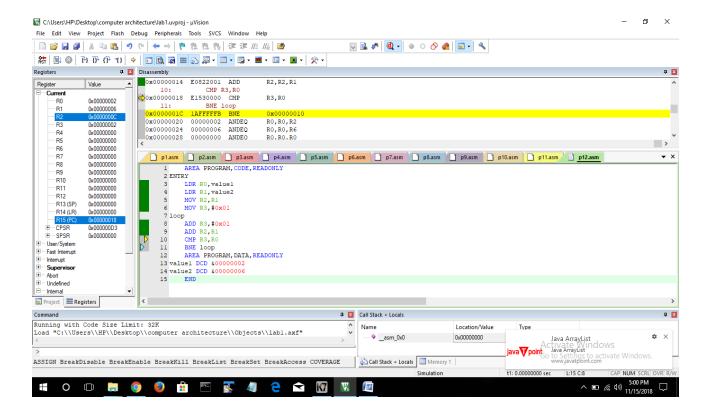
ENTRY

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 &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 RO,RO,R1

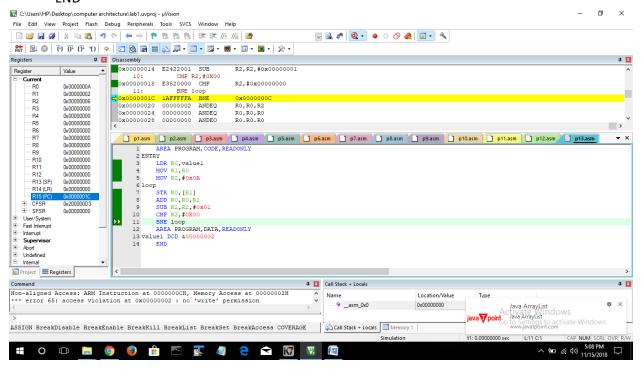
SUB R2,R2,#0x01

CMP R2,#0X00

BNE loop

AREA PROGRAM, DATA, READONLY

value1 DCD &00000002



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

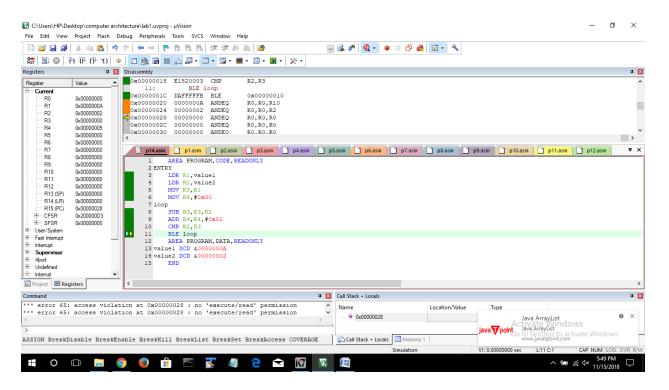
SUB R3,R3,R2

ADD R4 R4 #0x01
```

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

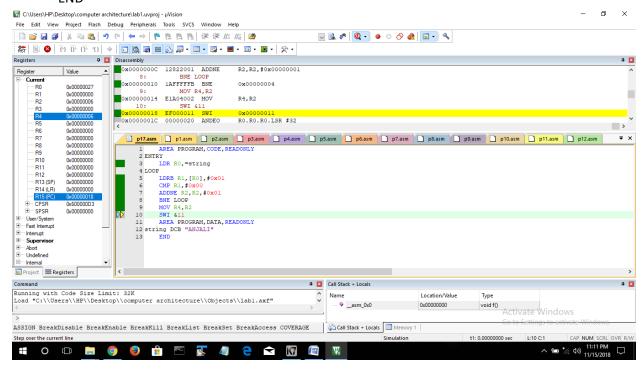
BNE LOOP

MOV R4,R2

SWI &11

AREA PROGRAM, DATA, READONLY

string DCB "ANJALI"



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:

0 🗆

AREA PROGRAM, CODE, READONLY **ENTRY** LDR RO,=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** 0 × File Edit View Project Flash Debug Peripherals Tools SVCS Window Help □ 😅 🖫 🗿 🐰 👊 选 👩 🥲 🖛 → | 🤁 数 数 数 課 課 准 版 🐸 # 🗵 Disassembly Value 0x00000027 R1 R2 R3 R4 R5 R6 R7 R8 R9 RO.RO.RO.LSR #32 0x00000000 p18.asm p1.asm p2.asm p3.asm p4.asm p5.asm p5.asm p6.asm p7.asm p8.asm p9.asm p10.asm p11.asm 0x00000000 0x00000000 1 AREA PROGRAM, CODE, READ 2 ENTRY 3 LDR R0, =string 0x00000000 0x00000000 R10
R11
R12
R13 (SP)
R14 (LR)
R15 (PC)
CPSR
SPSR
User/System
Fast Interrupt
Interrupt
Supervisor 0×00000000 4 LOOP
5 LDRB R1, [R0], \$0x01
6 CMP R1,"A"
7 ADDEO R2, R2, \$0x01
9 ENE LOOP
10 SWI 411
1 AREA PROGRAM, DATA, READONLY
12 string DCB "ANUALI"
3 END 4 T.OOP 0×00000000 User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal Internal Registers □ Call Stack + Locals ηХ Running with Code Size Limit: 32K Location/Value Load "C:\\Users\\HP\\Desktop\\computer architecture\\Objects\\labl.axf" void f0
Activate Windows 0.00000000 ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE

a e

₹ 17

へ ■ *// (小) 11:29 PM 11/15/2018

9. Write a program to add two integer strings.

CODE:

AREA PROGRAM, CODE, READONLY

ENTRY

LDR RO,=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

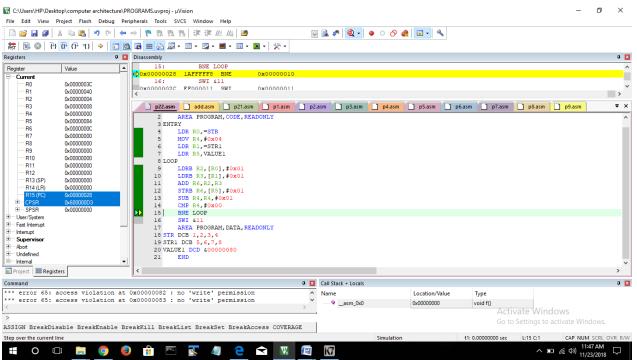
SWI &11

AREA PROGRAM, DATA, READONLY

STR DCB 1,2,3,4

STR1 DCB 5,6,7,8

VALUE1 DCD &00000080



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

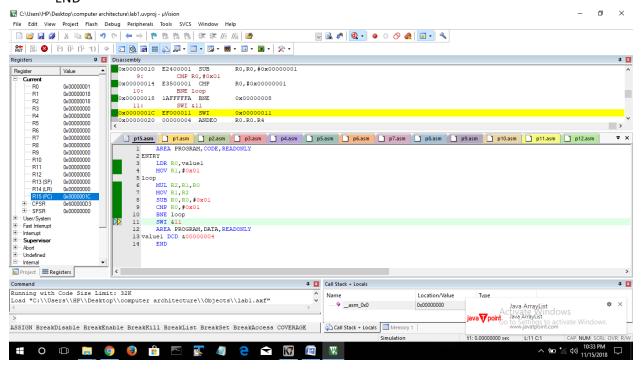
CMP R0,#0x01

BNE loop

SWI &11

AREA PROGRAM, DATA, READONLY

value1 DCD &00000004



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]

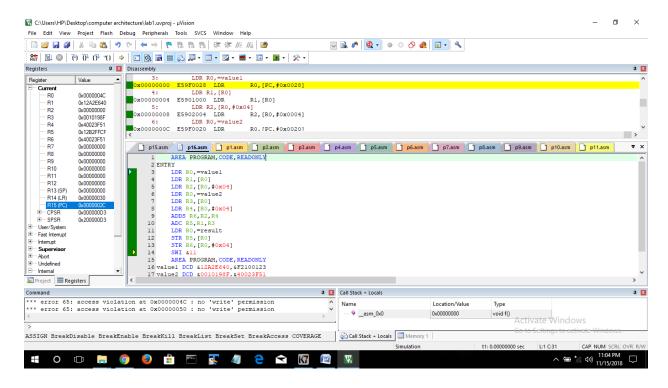
SWI &11

AREA PROGRAM, CODE, READONLY

value1 DCD &12A2E640,&F2100123

value2 DCD &0010198F,&40023F51

result DCD 0

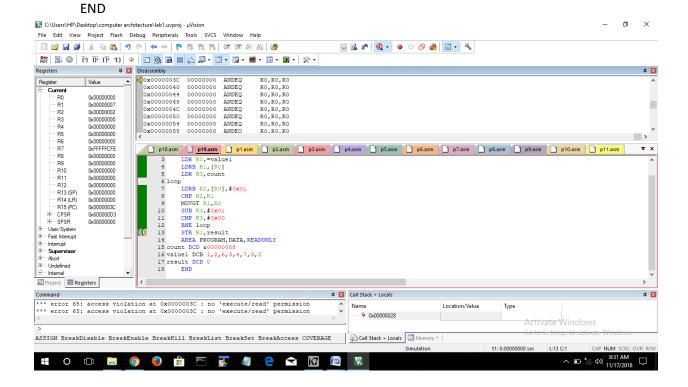


12. Write a program to find the largest number in an array.

CODE:

result DCD 0

```
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
```



13. Write a program to copy an array.

CODE:

AREA PROGRAM, CODE, READONLY

ENTRY

LDR RO,=STR

MOV R4,#0x04

LDR R5, VALUE1

LOOP

LDRB R2,[R0],#0x01

STRB R2,[R5],#0x01

SUB R4,R4,#0x01

CMP R4,#0x00

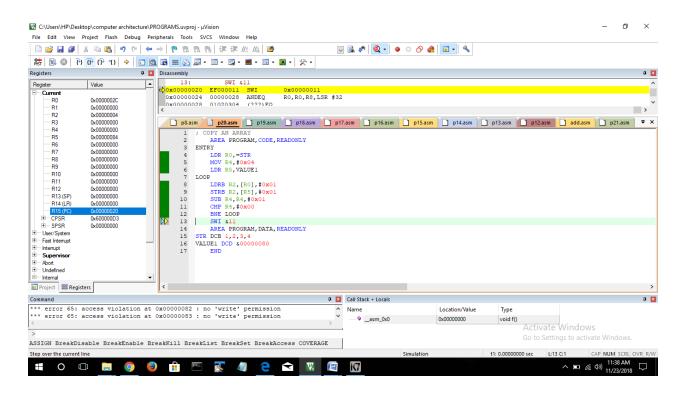
BNE LOOP

SWI &11

AREA PROGRAM, DATA, READONLY

STR DCB 1,2,3,4

VALUE1 DCD &00000080



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

i) ax2+by2

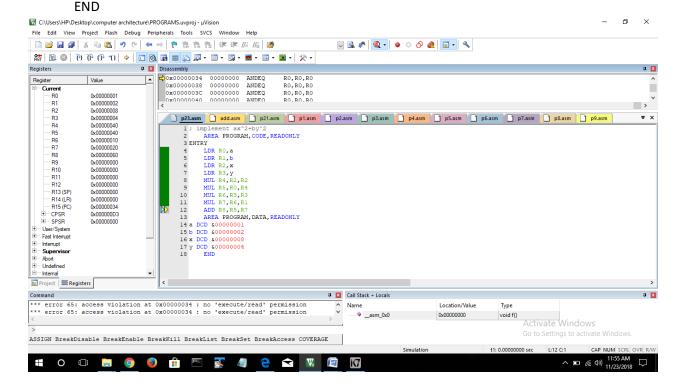
CODE:

```
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 R5,R0,R4
MUL R6,R3,R3
MUL R7,R6,R1
ADD R8,R5,R7
AREA PROGRAM,DATA,READONLY
a DCD &00000001
```

x DCD &00000008 y DCD &00000004

b DCD &00000002



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

CODE:

AREA PROGRAM, CODE, READONLY

ENTRY

LDR RO,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

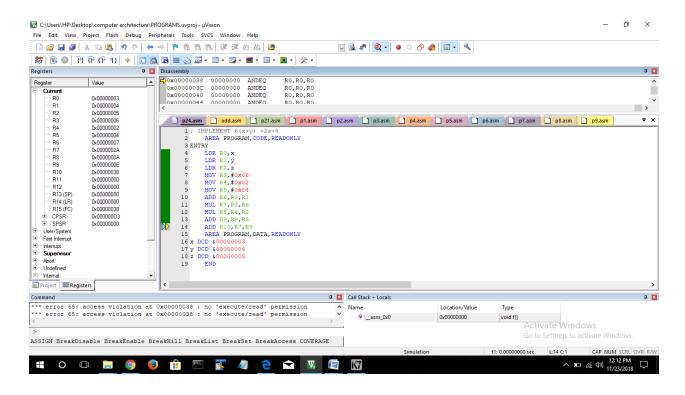
ADD R10,R7,R9

AREA PROGRAM, DATA, READONLY

x DCD &00000003

y DCD &00000004

z DCD &00000005



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

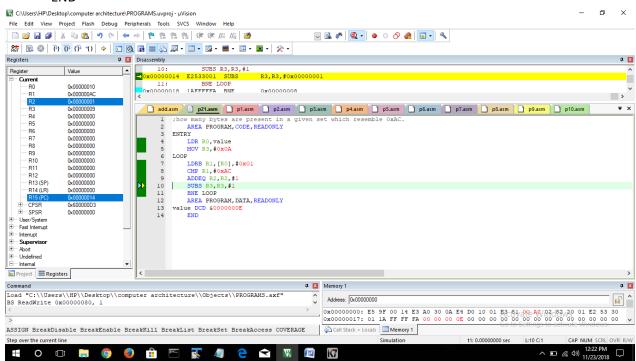
ADDEQ R2,R2,#1

SUBS R3,R3,#1

BNE LOOP

AREA PROGRAM,DATA,READONLY

value DCD &0000000E



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 RO, VALUE1

MOV R1,#0x20

MOV R2,#0x00

MOV R3,#0x00

LOOP

MOVS RO, RO, RRX

ADDCS R3,R3,#0x01

ADDCC R2,R2,#0x01

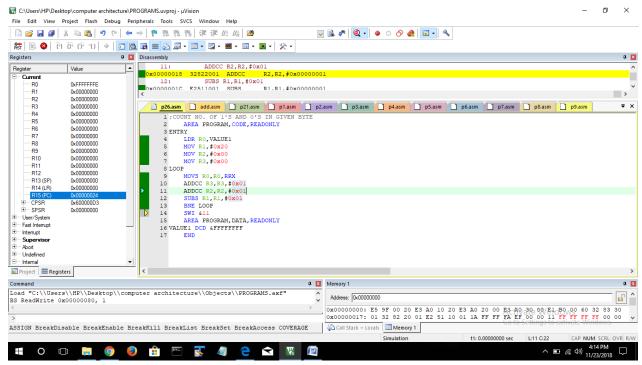
SUBS R1,R1,#0x01

BNE LOOP

SWI &11

AREA PROGRAM, DATA, READONLY

VALUE1 DCD &FFFFFFF



One's Comlement

CODE:

AREA PROGRAM, CODE, READONLY

ENTRY

LDR R1, value

MVN R2,R1

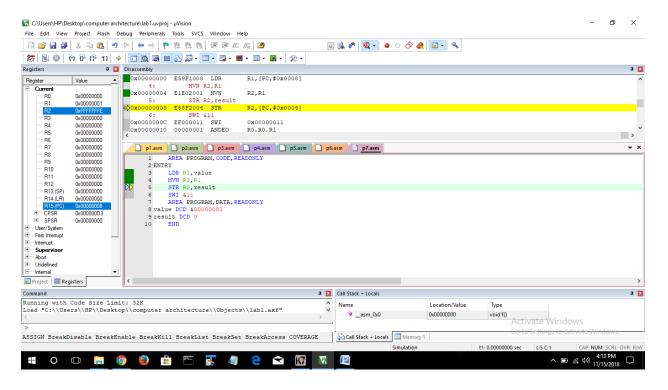
STR R2,result

SWI &11

AREA PROGRAM, DATA, READONLY

value DCD &0000001

result DCD 0



2's complement

CODE:

AREA PROGRAM, CODE, READONLY

ENTRY

LDR R1, value

MVN R2,R1

ADD R2,R2,#0x01

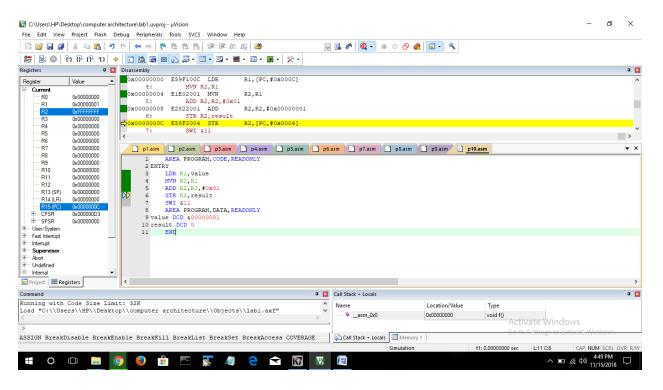
STR R2,result

SWI &11

AREA PROGRAM, DATA, READONLY

value DCD &0000001

result DCD 0



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
```

