Ayush Goyal

190905522 CSE D 62

ES Lab 3 (Week 3) Programs on Arithmetic and Logical Instructions

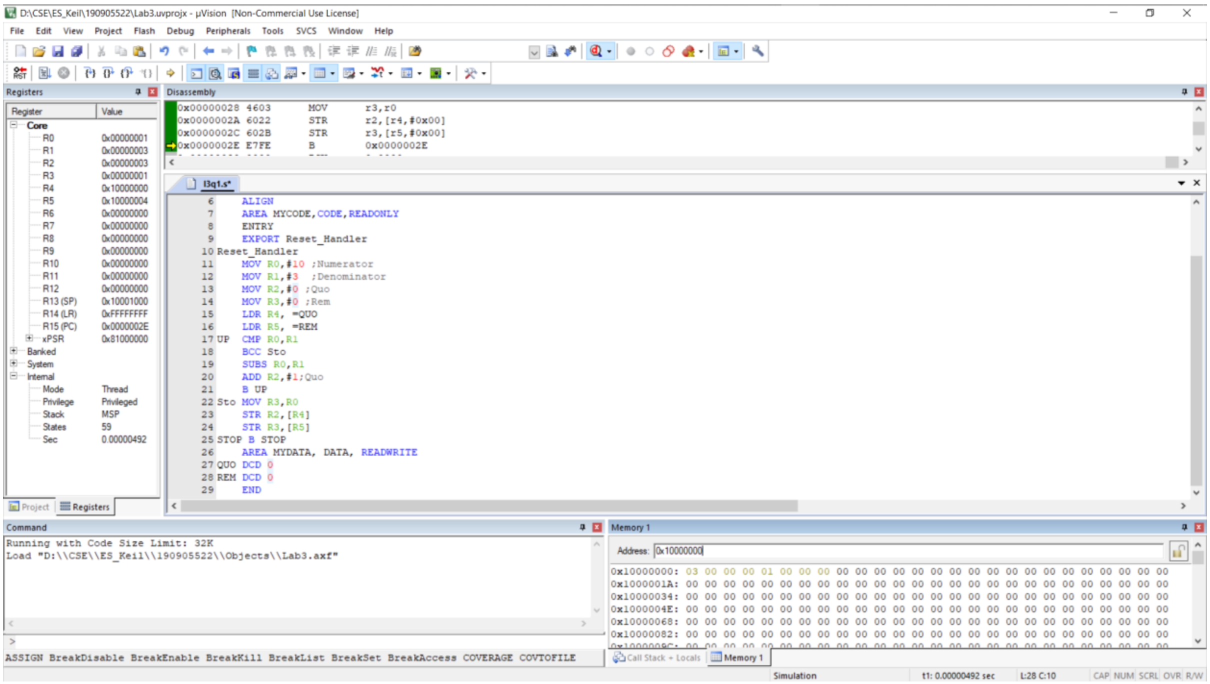
Q1) Write an assembly language program to implement division by repetitive subtraction.

CODE:

```
        AREA RESET,DATA,READONLY
        EXPORT __Vectors
__Vectors
        DCD 0X10001000
        DCD Reset_Handler
        ALIGN
        AREA MYCODE,CODE,READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
  MOV R0,#10
        MOV R1,#3
  MOV R2,#0 ;Quo
        MOV R3,#0 ;Rem
        LDR R4, =QUO
        LDR R5, =REM
UP      CMP R0,R1
        BCC Sto
        SUBS R0,R1
        ADD R2,#1;Quo
        B UP
Sto     MOV R3,R0
        STR R2,[R4]
        STR R3,[R5]
STOP B STOP
```

```
    AREA MYDATA, DATA, READWRITE
QUO DCD 0
REM DCD 0
            END
```

OUTPUT:



D:\CSE\ES_Keil\190905522\Lab3.uvprojx - µVision [Non-Commercial Use License]

Register	Value
Core	
R0	0x00000001
R1	0x00000003
R2	0x00000003
R3	0x00000001
R4	0x10000000
R5	0x10000004
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x0000002E
xPSR	0x81000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	59
Sec	0.00000492

```
0x00000028 4603      MOV      r3,r0
0x0000002A 6022      STR      r2,[r4,#0x00]
0x0000002C 602B      STR      r3,[r5,#0x00]
0x0000002E E7FE      B        0x0000002E
```

```
    ALIGN
    AREA MYCODE,CODE,READONLY
    ENTRY
    EXPORT Reset_Handler
Reset_Handler
    MOV R0,#10 ;Numerator
    MOV R1,#3  ;Denominator
    MOV R2,#0 ;Quo
    MOV R3,#0 ;Rem
    LDR R4, =QUO
    LDR R5, =REM
UP  CMP R0,R1
    BCC Sto
    SUBS R0,R1
    ADD R2,#1;Quo
    B UP
Sto MOV R3,R0
    STR R2,[R4]
    STR R3,[R5]
STOP B STOP
    AREA MYDATA, DATA, READWRITE
QUO DCD 0
REM DCD 0
    END
```

```
Running with Code Size Limit: 32K
Load "D:\\CSE\\ES_Keil\\190905522\\Objects\\Lab3.axf"
```

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Address: 0x10000000

```
0x10000000: 03 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000034: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000004E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000082: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000492 sec   L:28 C:10   CAP NUM SCRL OVR R/W

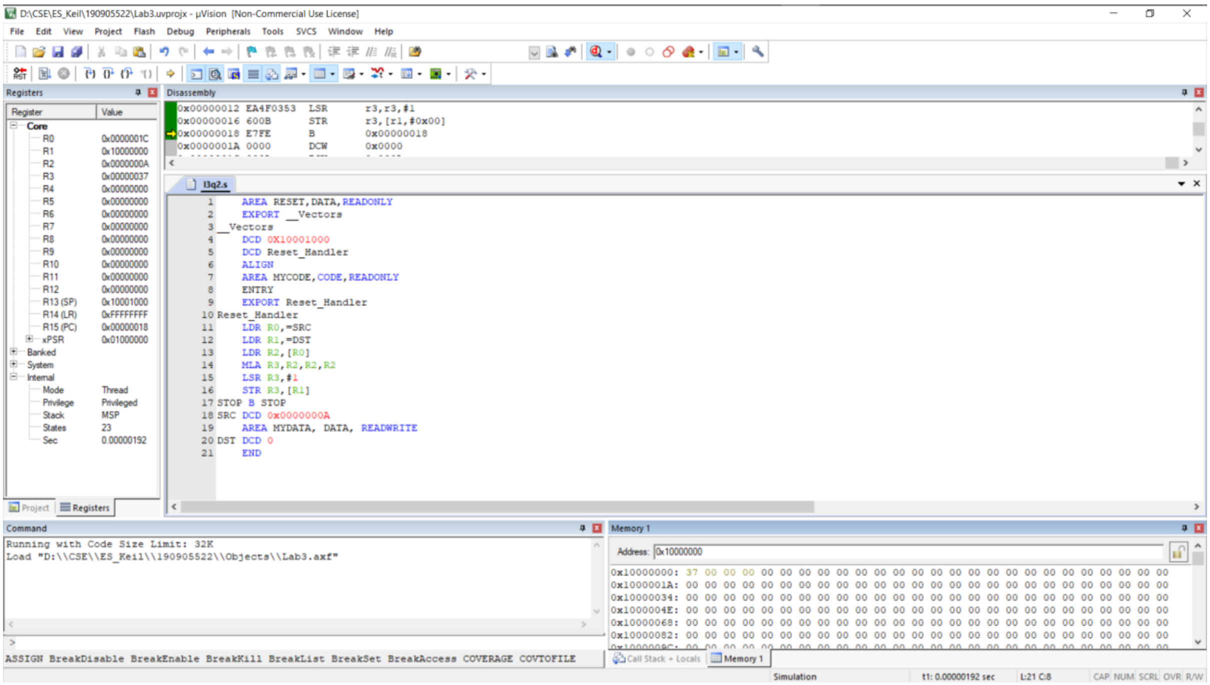
Q2) Find the sum of ‘n’ natural numbers using MLA instruction.

CODE:

```
            AREA RESET,DATA,READONLY
            EXPORT __Vectors
__Vectors
            DCD 0X10001000
            DCD Reset_Handler
            ALIGN
            AREA MYCODE,CODE,READONLY
            ENTRY
            EXPORT Reset_Handler
Reset_Handler
```

```
   LDR R0,=SRC
          LDR R1,=DST
          LDR R2,[R0]
          MLA R3,R2,R2,R2
          LSR R3,#1
          STR R3,[R1]
STOP B STOP
SRC DCD 0x0000000A
   AREA MYDATA, DATA, READWRITE
DST DCD 0
          END
```

OUTPUT:



D:\CSE\ES_Keil\190905522\Lab3.uvprojx - µVision [Non-Commercial Use License]

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
Core	
R0	0x0000001C
R1	0x10000000
R2	0x0000000A
R3	0x00000037
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000018
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	23
Sec	0.00000192

Disassembly

```
0x00000012 EA4F0353  LSR      r3,r3,#1
0x00000016 600B      STR      r3,[r1,#0x00]
0x00000018 E7FE      B        0x00000018
0x0000001A 0000      DCW      0x0000
```

l3q2.s

```
    AREA RESET,DATA,READONLY
    EXPORT __Vectors
__Vectors
    DCD 0X10001000
    DCD Reset_Handler
    ALIGN
    AREA MYCODE,CODE,READONLY
    ENTRY
    EXPORT Reset_Handler
Reset_Handler
    LDR R0,=SRC
    LDR R1,=DST
    LDR R2,[R0]
    MLA R3,R2,R2,R2
    LSR R3,#1
    STR R3,[R1]
STOP B STOP
SRC DCD 0x0000000A
    AREA MYDATA, DATA, READWRITE
DST DCD 0
    END
```

Command

```
Running with Code Size Limit: 32K
Load "D:\\CSE\\ES_Keil\\190905522\\Objects\\Lab3.axf"
```

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Memory 1

Address: 0x10000000

```
0x10000000: 37 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000034: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000004E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000082: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Q3) Write an assembly language program to find GCD and LCM of two 8-bit numbers.

CODE:

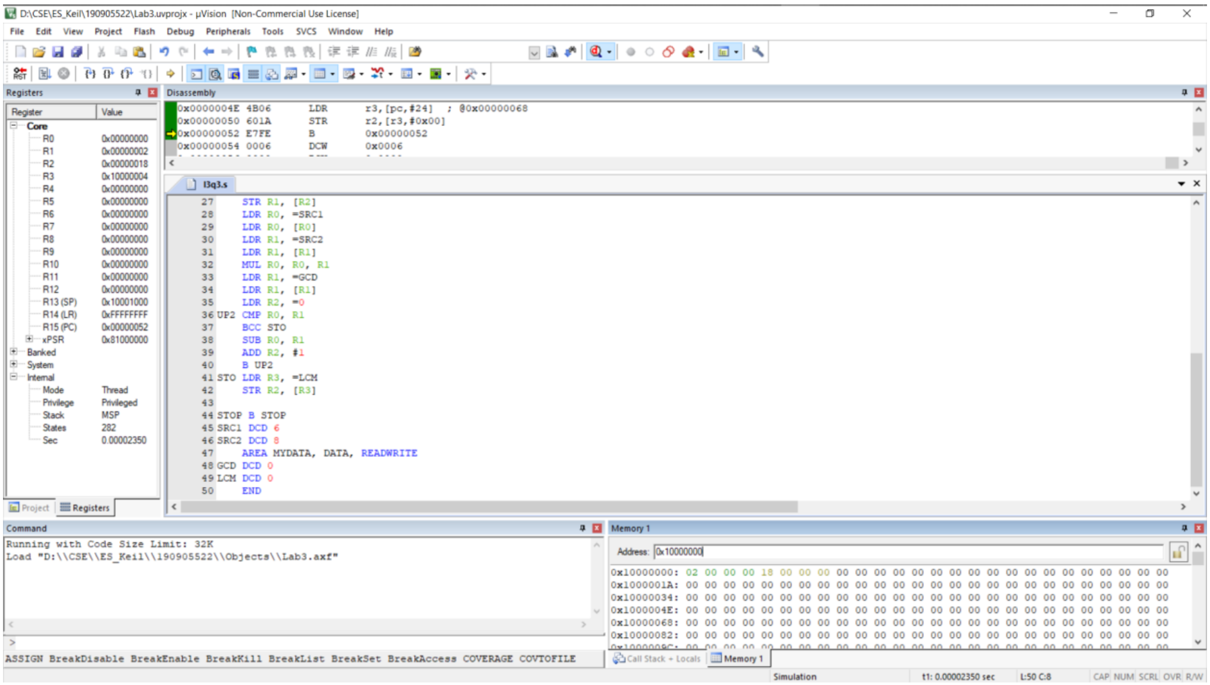
```
          AREA RESET,DATA,READONLY
                    EXPORT __Vectors
__Vectors
```

```
          DCD 0X10001000
          DCD Reset_Handler
          ALIGN
          AREA MYCODE,CODE,READONLY
          ENTRY
          EXPORT Reset_Handler
Reset_Handler
          LDR R0, =SRC1
          LDR R0, [R0]
          LDR R1, =SRC2
          LDR R1, [R1]
AGA       CMP R0, #0
          BEQ EXIT
UP        CMP R1, R0
          BHS CON
          B SKIP
CON       SUB R1, R1, R0
          B UP
SKIP MOV R2, R0
          MOV R0, R1
          MOV R1, R2
          B AGA
EXIT LDR R2, =GCD
          STR R1, [R2]
          LDR R0, =SRC1
          LDR R0, [R0]
          LDR R1, =SRC2
          LDR R1, [R1]
          MUL R0, R0, R1
          LDR R1, =GCD
          LDR R1, [R1]
```

```
        LDR R2, =0
UP2 CMP R0, R1
        BCC STO
        SUB R0, R1
        ADD R2, #1
        B UP2
STO LDR R3, =LCM
        STR R2, [R3]

STOP B STOP
SRC1 DCD 6
SRC2 DCD 8
        AREA MYDATA, DATA, READWRITE
GCD     DCD 0
LCM DCD 0
        END
```

OUTPUT:


D:\CSE\ES_Keil\190905522\Lab3.uvprojx - µVision [Non-Commercial Use License]
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
l3q3.s
Command
Memory 1
Project
Registers
Call Stack + Locals
Memory 1
Simulation
t1: 0.00002350 sec
L:50 C:8
CAP NUM SCRL OVR R/W

Register	Value
Core	
R0	0x00000000
R1	0x00000002
R2	0x00000018
R3	0x10000004
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000052
xPSR	0x81000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	282
Sec	0.00002350

```
0x0000004E 4B06      LDR      r3,[pc,#24]  ; @0x00000068
0x00000050 601A      STR      r2,[r3,#0x00]
0x00000052 E7FE      B        0x00000052
0x00000054 0006      DCW      0x0006
```

```
        STR R1, [R2]
        LDR R0, =SRC1
        LDR R0, [R0]
        LDR R1, =SRC2
        LDR R1, [R1]
        MUL R0, R0, R1
        LDR R1, =GCD
        LDR R1, [R1]
        LDR R2, =0
UP2 CMP R0, R1
        BCC STO
        SUB R0, R1
        ADD R2, #1
        B UP2
STO LDR R3, =LCM
        STR R2, [R3]

STOP B STOP
SRC1 DCD 6
SRC2 DCD 8
        AREA MYDATA, DATA, READWRITE
GCD DCD 0
LCM DCD 0
        END
```

```
Running with Code Size Limit: 32K
Load "D:\\CSE\\ES_Keil\\190905522\\Objects\\Lab3.axf"
```

```
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Address: 0x10000000

```
0x10000000: 02 00 00 00 18 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000034: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000004E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000082: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

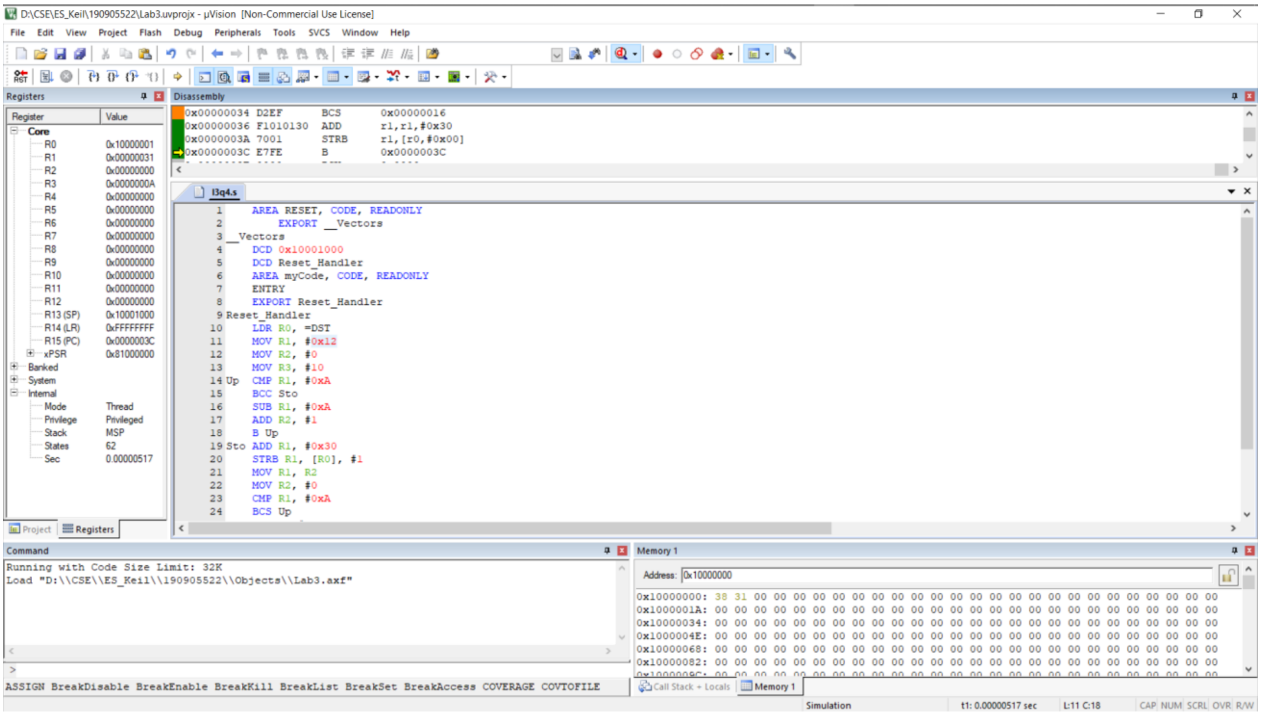
Q4) Write an ARM assembly language program to convert 2-digit hexadecimal number into ascii format.

CODE:

```
        AREA RESET, CODE, READONLY
                EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler
        AREA myCode, CODE, READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0, =DST
        MOV R1, #0x12A
        MOV R2, #0
        MOV R3, #10
Up      CMP R1, #0xA
        BCC Sto
        SUB R1, #0xA
        ADD R2, #1
        B Up
Sto     ADD R1, #0x30
        STRB R1, [R0], #1
        MOV R1, R2
        MOV R2, #0
        CMP R1, #0xA
        BCS     Up
        ADD R1,#0x30
        STRB R1,[R0]
STOP B STOP
        AREA MYDATA, DATA, READWRITE
```

```
DST DCD 0x0
        END
```

OUTPUT:



D:\CSE\ES_Keil\190905522\Lab3.uvprojx - µVision [Non-Commercial Use License]

```
Register      Value
Core
  R0          0x10000001
  R1          0x00000031
  R2          0x00000000
  R3          0x0000000A
  R4          0x00000000
  R5          0x00000000
  R6          0x00000000
  R7          0x00000000
  R8          0x00000000
  R9          0x00000000
  R10         0x00000000
  R11         0x00000000
  R12         0x00000000
  R13 (SP)    0x10001000
  R14 (LR)    0xFFFFFFFF
  R15 (PC)    0x0000003C
  xPSR        0x81000000
Banked
System
Internal
  Mode        Thread
  Privilege   Privileged
  Stack       MSP
  States      62
  Sec         0.00000517
```

```
0x00000034 D2EF      BCS      0x00000016
0x00000036 F1010130  ADD      r1,r1,#0x30
0x0000003A 7001      STRB     r1,[r0,#0x00]
0x0000003C E7FE      B        0x0000003C
```

l3q4.s

```
        AREA RESET, CODE, READONLY
            EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler
        AREA myCode, CODE, READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0, =DST
        MOV R1, #0x12
        MOV R2, #0
        MOV R3, #10
Up      CMP R1, #0xA
        BCC Sto
        SUB R1, #0xA
        ADD R2, #1
        B Up
Sto     ADD R1, #0x30
        STRB R1, [R0], #1
        MOV R1, R2
        MOV R2, #0
        CMP R1, #0xA
        BCS Up
```

```
Running with Code Size Limit: 32K
Load "D:\\CSE\\ES_Keil\\190905522\\Objects\\Lab3.axf"
```

```
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Address: 0x10000000

```
0x10000000: 38 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000034: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000004E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000082: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000517 sec   L:11 C:18   CAP NUM SCRL OVR R/W

Q5) Write an ARM assembly language program to convert a 32-bit BCD number in the unpacked form into packed form.

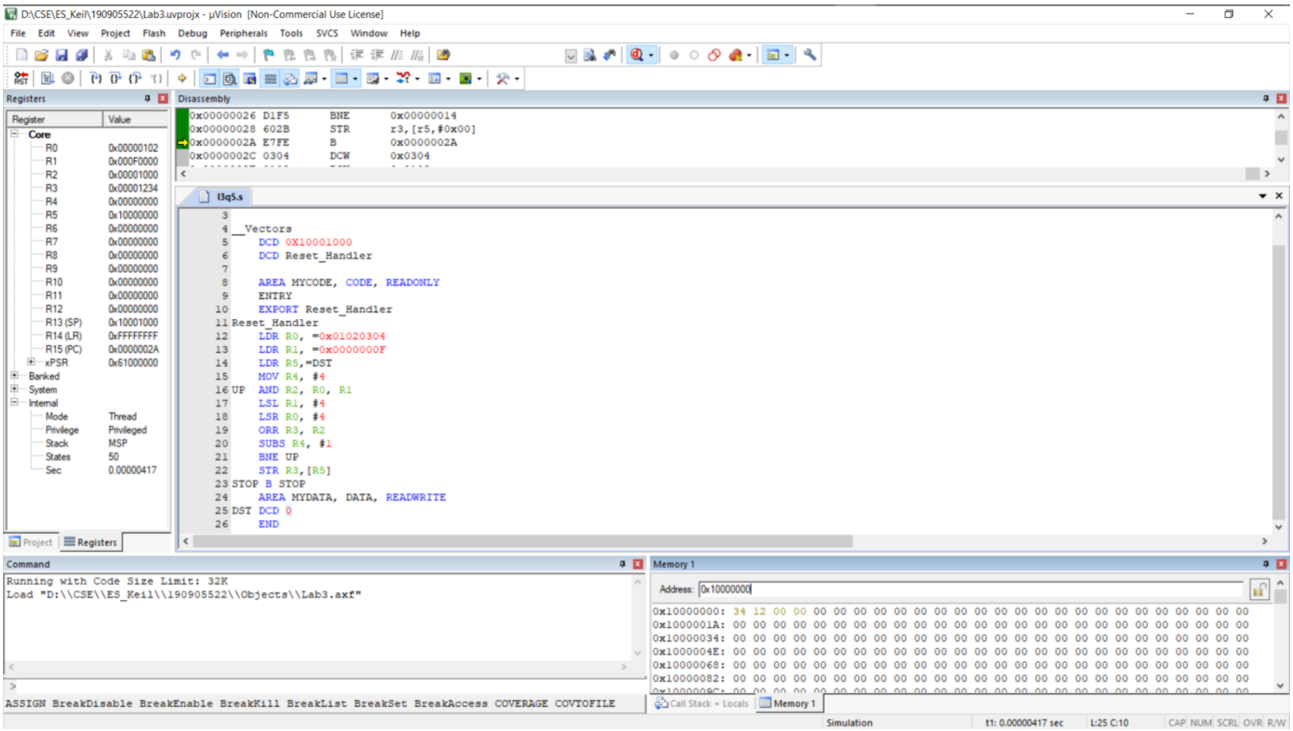
CODE:

```
        AREA RESET, DATA, READONLY
        EXPORT __Vectors


__Vectors
        DCD 0X10001000
        DCD Reset_Handler


        AREA MYCODE, CODE, READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0, =0x01020304
        LDR R1, =0x0000000F
        LDR R5,=DST
        MOV R4, #4
UP      AND R2, R0, R1
        LSL R1, #4
        LSR R0, #4
        ORR R3, R2
        SUBS R4, #1
        BNE UP
        STR R3,[R5]
STOP B STOP
        AREA MYDATA, DATA, READWRITE
DST DCD 0
        END
```

OUTPUT:



D:\CSE\ES_Keil\190905522\Lab3.uvprojx - µVision [Non-Commercial Use License]

Register	Value
Core	
R0	0x00000102
R1	0x000F0000
R2	0x00001000
R3	0x00001234
R4	0x00000000
R5	0x10000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x0000002A
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	50
Sec	0.00000417

```
0x00000026 D1F5      BNE      0x00000014
0x00000028 602B      STR      r3,[r5,#0x00]
0x0000002A E7FE      B        0x0000002A
0x0000002C 0304      DCW      0x0304
```

```

__Vectors
        DCD 0X10001000
        DCD Reset_Handler

        AREA MYCODE, CODE, READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0, =0x01020304
        LDR R1, =0x0000000F
        LDR R5,=DST
        MOV R4, #4
UP      AND R2, R0, R1
        LSL R1, #4
        LSR R0, #4
        ORR R3, R2
        SUBS R4, #1
        BNE UP
        STR R3,[R5]
STOP B STOP
        AREA MYDATA, DATA, READWRITE
DST DCD 0
        END
```

```
Running with Code Size Limit: 32K
Load "D:\\CSE\\ES_Keil\\190905522\\Objects\\Lab3.axf"
>
ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE COVTOFILE
```

Address: 0x10000000

```
0x10000000: 34 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000001A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000034: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000004E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000068: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000082: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Simulation   t1: 0.00000417 sec   L:25 C:10   CAP NUM SCRL OVR R/W

THE END