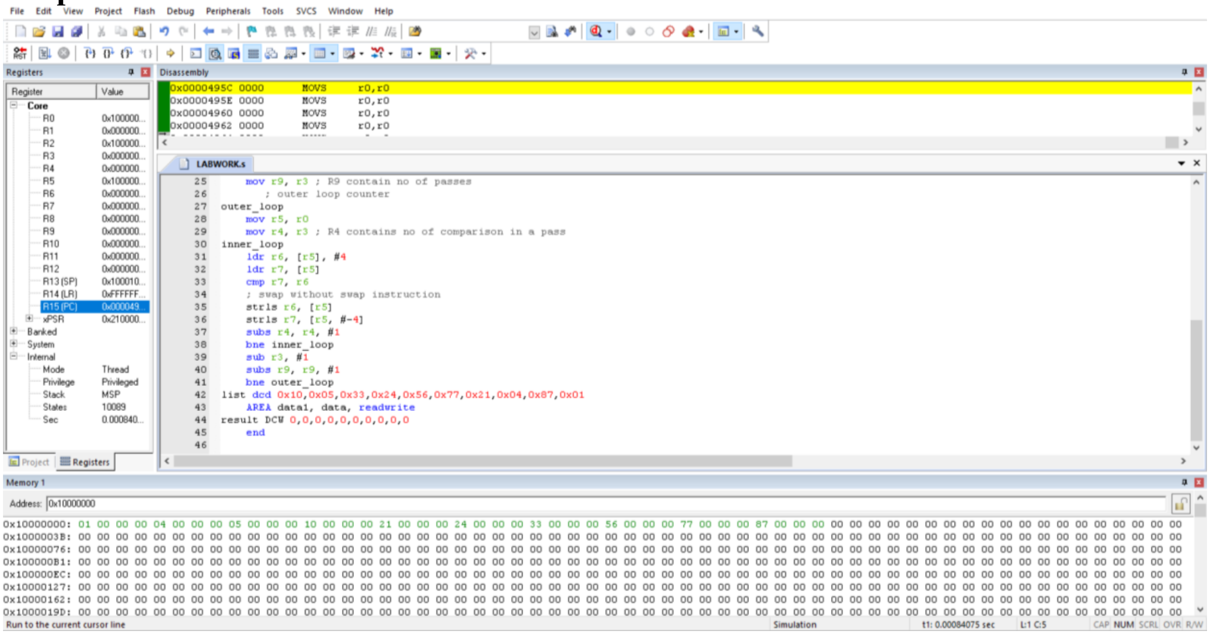
Solved Exercise:

Write an ARM ALP to sort a list using bubble sort.

```
        AREA RESET,DATA,READONLY
        EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler
        ALIGN
        AREA mycode,CODE,READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        mov r4,#0
        mov r1,#10
        ldr r0, =list
        ldr r2, =result
up      ldr r3, [r0,r4]
        str r3, [r2,r4]
        add r4, #04
        sub r1,#01
        cmp r1,#00
        bhi up
        ldr r0, =result
        mov r3, #10 ; inner loop counter
        sub r3, r3, #1
        mov r9, r3 ; R9 contain no of passes
    ; outer loop counter
outer_loop
        mov r5, r0
        mov r4, r3 ; R4 contains no of comparison in a pass
inner_loop
        ldr r6, [r5], #4
        ldr r7, [r5]
        cmp r7, r6
        ; swap without swap instruction
        strls r6, [r5]
        strls r7, [r5, #-4]
        subs r4, r4, #1
        bne inner_loop
        sub r3, #1
        subs r9, r9, #1
        bne outer_loop
list dcd 0x10,0x05,0x33,0x24,0x56,0x77,0x21,0x04,0x87,0x01
        AREA data1, data, readwrite
```

```
result DCW 0,0,0,0,0,0,0,0,0,0
        end
```

Output:



File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Disassembly
LABWORK.s
Project
Registers
Memory 1

Register	Value
Core	
R0	0x100000...
R1	0x000000...
R2	0x100000...
R3	0x000000...
R4	0x000000...
R5	0x100000...
R6	0x000000...
R7	0x000000...
R8	0x000000...
R9	0x000000...
R10	0x000000...
R11	0x000000...
R12	0x000000...
R13 (SP)	0x100010...
R14 (LR)	0xFFFFFF...
R15 (PC)	0x000049...
xPSR	0x210000...
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	10089
Sec	0.000840...

```
0x0000495C 0000      MOVS     r0,r0
0x0000495E 0000      MOVS     r0,r0
0x00004960 0000      MOVS     r0,r0
0x00004962 0000      MOVS     r0,r0
```

```
    mov r9, r3 ; R9 contain no of passes
       ; outer loop counter
outer_loop
    mov r5, r0
    mov r4, r3 ; R4 contains no of comparison in a pass
inner_loop
    ldr r6, [r5], #4
    ldr r7, [r5]
    cmp r7, r6
    ; swap without swap instruction
    strls r6, [r5]
    strls r7, [r5, #-4]
    subs r4, r4, #1
    bne inner_loop
    sub r3, #1
    subs r9, r9, #1
    bne outer_loop
list dcd 0x10,0x05,0x33,0x24,0x56,0x77,0x21,0x04,0x87,0x01
    AREA data1, data, readwrite
result DCW 0,0,0,0,0,0,0,0,0,0
    end
```

Address: 0x10000000

```
0x10000000: 01 00 00 00 04 00 00 00 05 00 00 00 10 00 00 00 21 00 00 00 24 00 00 00 33 00 00 00 56 00 00 00 77 00 00 00 87 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000003B: 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 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 00 00 00 00 00 00 00
0x10000076: 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 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 00 00 00 00 00 00 00
0x100000B1: 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 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 00 00 00 00 00 00 00
0x100000EC: 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 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 00 00 00 00 00 00 00
0x10000127: 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 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 00 00 00 00 00 00 00
0x10000162: 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 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 00 00 00 00 00 00 00
0x1000019D: 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 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 00 00 00 00 00 00 00
```

Run to the current cursor line   Simulation   t1: 0.00084075 sec   L:1 C:5   CAP NUM SCRL OVR R/W

Lab Exercises:

1. Write an assembly program to sort an array using selection sort.

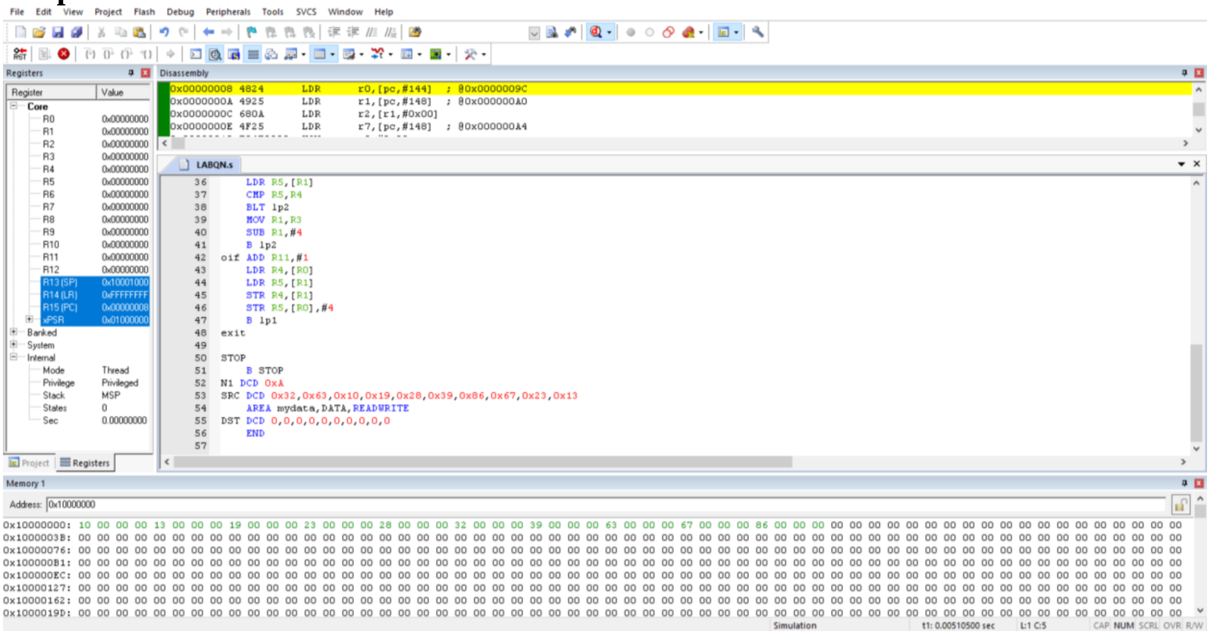
Program:

```
        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        ;r0 is pointer to ith element

        LDR R1, =N1

        LDR R2,[r1]         ;r2 stores number of elements

        LDR R7, =DST

        MOV R8,#0

up      CMP R8,R2
```

```
        BEQ out
        ADD R8,#1
        LDR R9,[R0],#4
        STR R9,[R7],#4
        B up
out     LDR R0,=DST
        MOV R1, R0          ;r1 is pointer to element to swap
        MOV R3,R0           ;r3 is pointer to jth element
        MOV R10,#0          ;r10 is counter for inner(j) loop
        MOV R11,#0          ;r11 is counter for outer(i) loop
lp1     CMP R11, R2         ;comparing i<10
        BEQ exit
        ADD R3,R0,#4        ;sets jth pointer to A[i+1]
        MOV R1,R0           ;sets swap element to A[i]
        ADD R10,R11,#1      ;j=i+1
lp2     CMP R10,R2          ;j<10
        BEQ oif
        ADD R10,#1          ;j++
        LDR R4,[R3],#4
        LDR R5,[R1]
        CMP R5,R4
        BLT lp2
        MOV R1,R3
        SUB R1,#4
        B lp2
oif     ADD R11,#1
        LDR R4,[R0]
        LDR R5,[R1]
        STR R4,[R1]
        STR R5,[R0],#4
        B lp1
```

```
exit


STOP
        B STOP
N1 DCD 0xA
SRC DCD 0x32,0x63,0x10,0x19,0x28,0x39,0x86,0x67,0x23,0x13
        AREA mydata,DATA,READWRITE
DST DCD 0,0,0,0,0,0,0,0,0,0
        END
```

Output:



Register	Value
Core	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
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)	0x00000008
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	0
Sec	0.00000000

```
0x00000008 4824      LDR      r0,[pc,#144]  ; @0x0000009C
0x0000000A 4925      LDR      r1,[pc,#148]  ; @0x000000A0
0x0000000C 680A      LDR      r2,[r1,#0x00]
0x0000000E 4F25      LDR      r7,[pc,#148]  ; @0x000000A4
```

```
        LDR R5,[R1]
        CMP R5,R4
        BLT lp2
        MOV R1,R3
        SUB R1,#4
        B lp2
oif ADD R11,#1
        LDR R4,[R0]
        LDR R5,[R1]
        STR R4,[R1]
        STR R5,[R0],#4
        B lp1
exit

STOP
        B STOP
N1 DCD 0xA
SRC DCD 0x32,0x63,0x10,0x19,0x28,0x39,0x86,0x67,0x23,0x13
        AREA mydata,DATA,READWRITE
DST DCD 0,0,0,0,0,0,0,0,0,0
        END
```

Address: 0x10000000

```
0x10000000: 10 00 00 00 13 00 00 00 19 00 00 00 23 00 00 00 28 00 00 00 32 00 00 00 39 00 00 00 63 00 00 00 67 00 00 00 86 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000003B: 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 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 00 00 00 00 00 00 00
0x10000076: 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 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 00 00 00 00 00 00 00
0x100000B1: 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 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 00 00 00 00 00 00 00
0x100000EC: 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 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 00 00 00 00 00 00 00
0x10000127: 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 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 00 00 00 00 00 00 00
0x10000162: 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 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 00 00 00 00 00 00 00
0x1000019D: 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 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 00 00 00 00 00 00 00
```

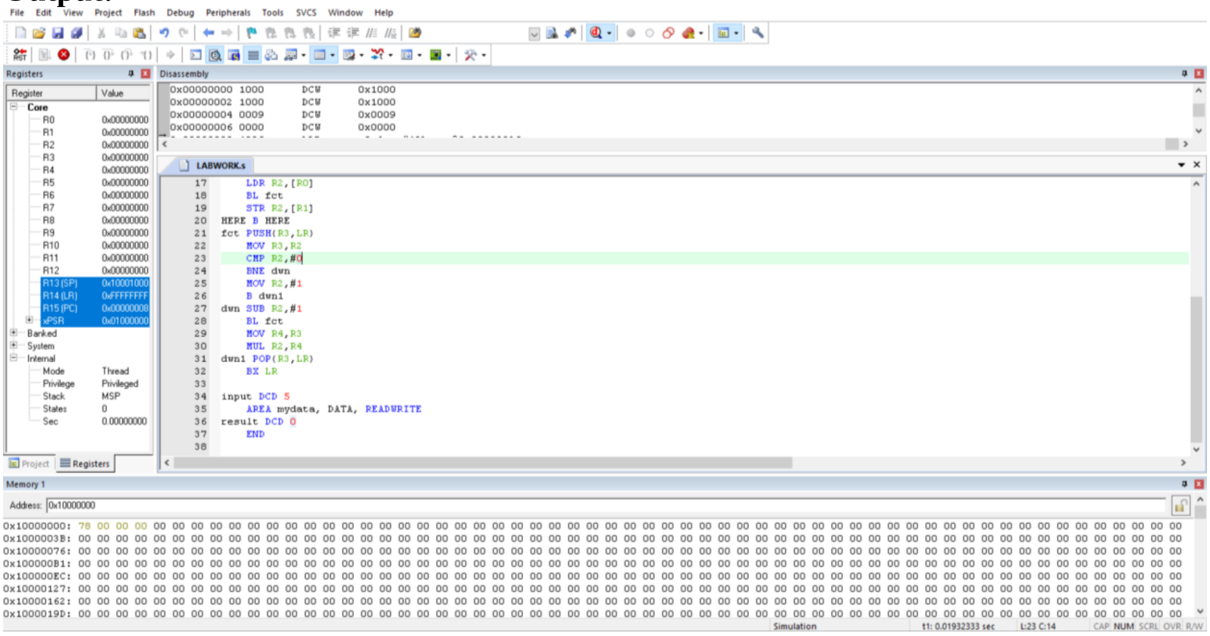
2. Write an assembly program to find the factorial of an unsigned number using recursion

Program:

```
        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,=input
        LDR R1,=result
        LDR R2,[R0]
        BL fct
        STR R2,[R1]
HERE B HERE
fct PUSH{R3,LR}
        MOV R3,R2
        CMP R2,#0
        BNE dwn
        MOV R2,#1
        B dwn1
dwn SUB R2,#1
        BL fct
        MOV R4,R3
        MUL R2,R4
dwn1 POP{R3,LR}
        BX LR

input DCD 5
        AREA mydata, DATA, READWRITE
result DCD 0
        END
```

Output:



Register	Value
Core	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
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)	0x00000008
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	0
Sec	0.00000000

```
0x00000000 1000      DCW      0x1000
0x00000002 1000      DCW      0x1000
0x00000004 0009      DCW      0x0009
0x00000006 0000      DCW      0x0000
```

```
    LDR R2,[R0]
    BL fct
    STR R2,[R1]
HERE B HERE
fct PUSH(R3,LR)
    MOV R3,R2
    CMP R2,#0
    BNE dwn
    MOV R2,#1
    B dwn1
dwn SUB R2,#1
    BL fct
    MOV R4,R3
    MUL R2,R4
dwn1 POP(R3,LR)
    BX LR

input DCD 5
    AREA mydata, DATA, READWRITE
result DCD 0
    END
```

Address: 0x10000000

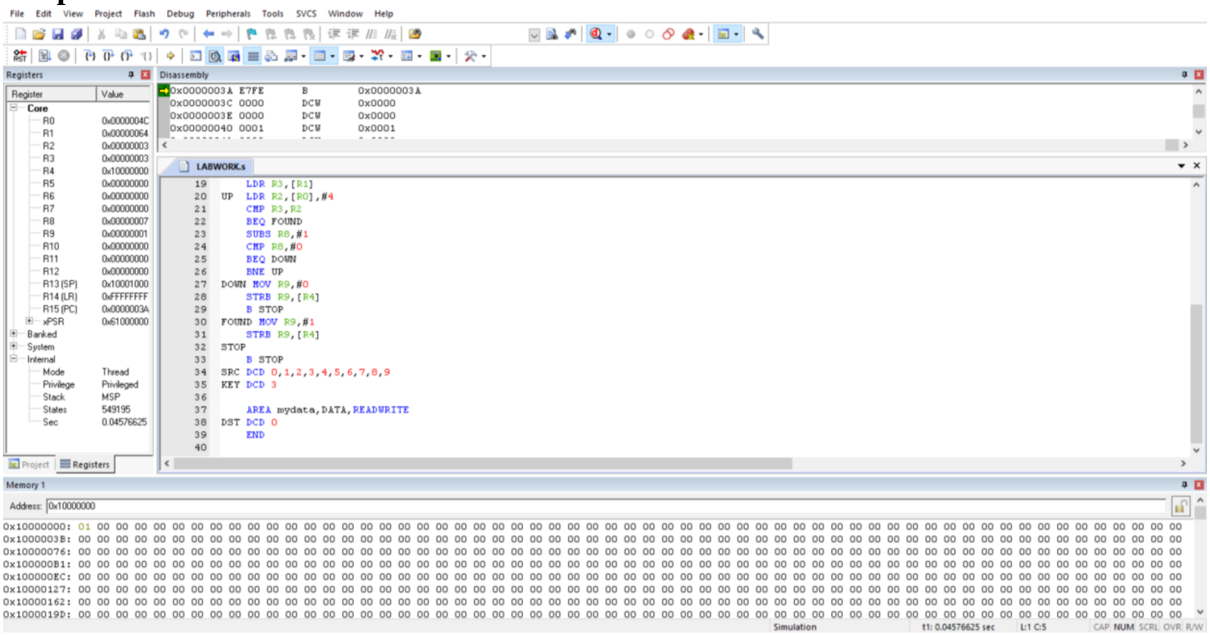
```
0x10000000: 78 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x1000003B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x10000076: 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 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 00 00 00 00 00 00 00
0x100000B1: 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 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 00 00 00 00 00 00 00
0x100000EC: 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 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 00 00 00 00 00 00 00
0x10000127: 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 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 00 00 00 00 00 00 00
0x10000162: 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 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 00 00 00 00 00 00 00
0x1000019D: 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 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 00 00 00 00 00 00 00
```

3. Write an assembly program to search an element in an array of ten 32 bit numbers using linear search.

Program:

```
        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,=KEY
        LDR R4,=DST
        MOV R8,#10
        LDR R3,[R1]
UP      LDR R2,[R0],#4
        CMP R3,R2
        BEQ FOUND
        SUBS R8,#1
        CMP R8,#0
        BEQ DOWN
        BNE UP
DOWN MOV R9,#0
        STRB R9,[R4]
        B STOP
FOUND MOV R9,#1
        STRB R9,[R4]
STOP
        B STOP
SRC DCD 0,1,2,3,4,5,6,7,8,9
KEY DCD 3

        AREA mydata,DATA,READWRITE
DST DCD 0
        END
```

Output:



Register	Value
Core	
R0	0x0000004C
R1	0x00000064
R2	0x00000003
R3	0x00000003
R4	0x10000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000007
R9	0x00000001
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x0000003A
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	549195
Sec	0.04576625

```
0x0000003A E7FE      B        0x0000003A
0x0000003C 0000      DCW      0x0000
0x0000003E 0000      DCW      0x0000
0x00000040 0001      DCW      0x0001
```

```
        LDR R3,[R1]
UP      LDR R2,[R0],#4
        CMP R3,R2
        BEQ FOUND
        SUBS R8,#1
        CMP R8,#0
        BEQ DOWN
        BNE UP
DOWN    MOV R9,#0
        STRB R9,[R4]
        B STOP
FOUND   MOV R9,#1
        STRB R9,[R4]
STOP
        B STOP
SRC DCD 0,1,2,3,4,5,6,7,8,9
KEY DCD 3

        AREA mydata,DATA,READWRITE
DST DCD 0
        END

```

```
Address: 0x10000000
0x10000000: 01 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 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 00 00 00 00 00 00
0x1000003B: 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 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 00 00 00 00 00 00 00
0x10000076: 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 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 00 00 00 00 00 00 00
0x100000B1: 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 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 00 00 00 00 00 00 00
0x100000EC: 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 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 00 00 00 00 00 00 00
0x10000127: 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 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 00 00 00 00 00 00 00
0x10000162: 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 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 00 00 00 00 00 00 00
0x1000019D: 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 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 00 00 00 00 00 00 00
```

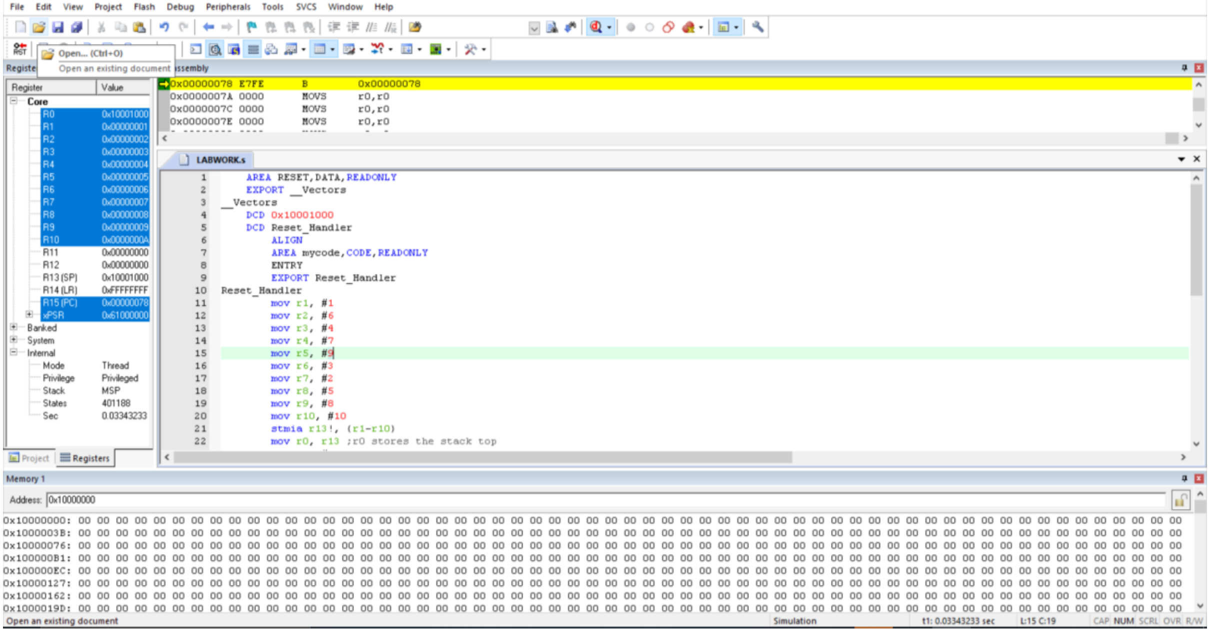
4. Assume that ten 32 bit numbers are stored in registers R1-R10. Sort these numbers in the empty ascending stack using selection sort and store the sorted array back into the registers. Use STM and LDMDB instructions wherever necessary.

Program:

```
          AREA RESET,DATA,READONLY

          EXPORT __Vectors

__Vectors

          DCD 0x10001000

          DCD Reset_Handler

                    ALIGN

                    AREA mycode,CODE,READONLY

                    ENTRY

                    EXPORT Reset_Handler

Reset_Handler

                    mov r1, #1

                    mov r2, #6

                    mov r3, #4

                    mov r4, #7

                    mov r5, #9

                    mov r6, #3

                    mov r7, #2
```

```
                mov r8, #5
                mov r9, #8
                mov r10, #10
                stmia r13!, {r1-r10}
                mov r0, r13 ;r0 stores the stack top
                mov r2, #10 ;r2 stores number of elements in stack
                mov r8,#0       ;r8 is counter for outer loop
ol              cmp r8,r2
                beq exit
                mov r1, r0
                mov r3, r0
                sub r3, #4
                add r9,r8,#1
il              cmp r9,r2
                beq exin
                add r9,#1
                ldmdb r1,{r4}
                ldmdb r3!,{r5}
                cmp r5,r4
                blt il
                stmdb r1,{r5}
                stm r3,{r4}
                b il
exin            sub r0,#4
                add r8,#1
                b ol
exit
                ldmdb r13!,{r1-r10}
stop            B stop
                AREA mydata,DATA,READWRITE
                END
```

Output:


File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Open... (Ctrl+O)
Open an existing document
Registe
ssembly
LABWORK.s
Project
Registers
Memory 1

Register	Value
Core	
R0	0x10001000
R1	0x00000001
R2	0x00000002
R3	0x00000003
R4	0x00000004
R5	0x00000005
R6	0x00000006
R7	0x00000007
R8	0x00000008
R9	0x00000009
R10	0x0000000A
R11	0x00000000
R12	0x00000000
R13 (SP)	0x10001000
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x00000078
xPSR	0x61000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	401188
Sec	0.03343233

```
0x00000078 E7FE      B        0x00000078
0x0000007A 0000      MOVS     r0,r0
0x0000007C 0000      MOVS     r0,r0
0x0000007E 0000      MOVS     r0,r0
```

```
    AREA RESET,DATA,READONLY
    EXPORT __Vectors
__Vectors
    DCD 0x10001000
    DCD Reset_Handler
        ALIGN
        AREA mycode,CODE,READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        mov r1, #1
        mov r2, #6
        mov r3, #4
        mov r4, #7
        mov r5, #9
        mov r6, #3
        mov r7, #2
        mov r8, #5
        mov r9, #8
        mov r10, #10
        stmia r13!, (r1-r10)
        mov r0, r13 ;r0 stores the stack top
```

Address: 0x10000000

```
0x10000000: 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 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 00 00 00 00 00 00 00
0x1000003B: 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 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 00 00 00 00 00 00 00
0x10000076: 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 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 00 00 00 00 00 00 00
0x100000B1: 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 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 00 00 00 00 00 00 00
0x100000EC: 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 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 00 00 00 00 00 00 00
0x10000127: 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 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 00 00 00 00 00 00 00
0x10000162: 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 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 00 00 00 00 00 00 00
0x1000019D: 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 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 00 00 00 00 00 00 00
```

Open an existing document | Simulation | t1: 0.03343233 sec | L:15 C:19 | CAP NUM SCRL OVR R/W