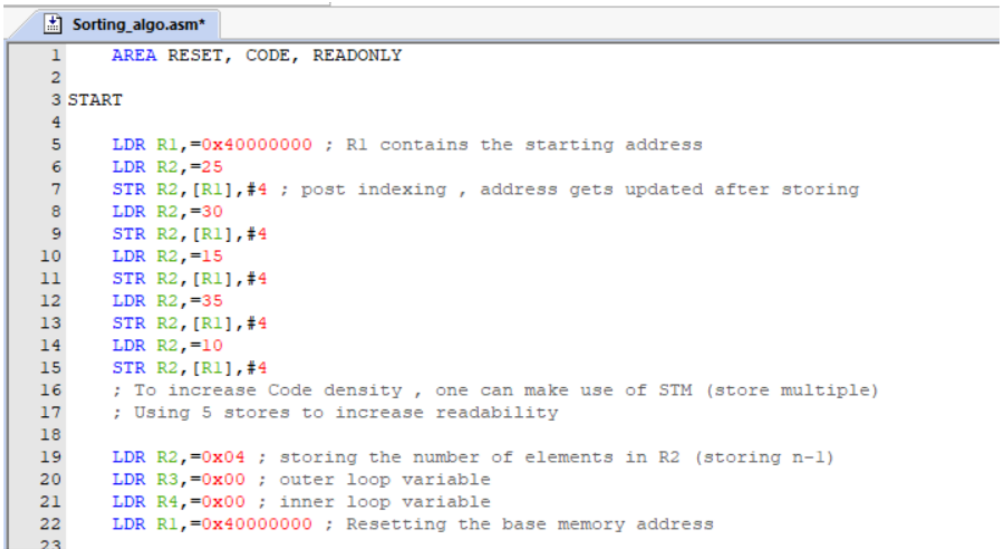
Name : Sai Rajat Goparaju **ID Number :** 2020A3PS0258H

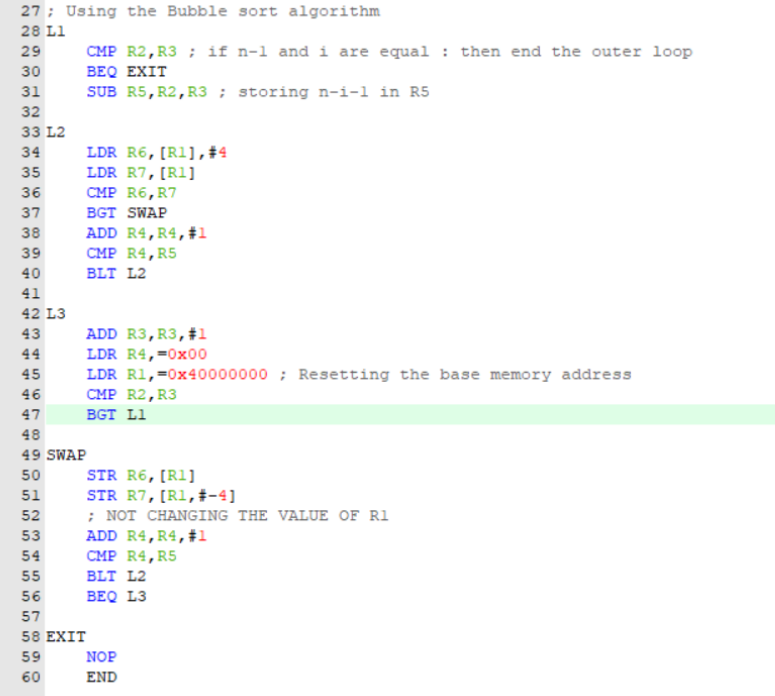
Documentation for the code:

Step 1: Load the numbers into the memory locations in an unsorted fashion. Using a variable , store thee number of elements to be sorted. In the program written , 5 numbers have been considered , and have been stored in the memory as follows : [25,30,15,35,10]



```
    AREA RESET, CODE, READONLY

START

    LDR R1,=0x40000000 ; R1 contains the starting address
    LDR R2,=25
    STR R2,[R1],#4 ; post indexing , address gets updated after storing
    LDR R2,=30
    STR R2,[R1],#4
    LDR R2,=15
    STR R2,[R1],#4
    LDR R2,=35
    STR R2,[R1],#4
    LDR R2,=10
    STR R2,[R1],#4
    ; To increase Code density , one can make use of STM (store multiple)
    ; Using 5 stores to increase readability

    LDR R2,=0x04 ; storing the number of elements in R2 (storing n-1)
    LDR R3,=0x00 ; outer loop variable
    LDR R4,=0x00 ; inner loop variable
    LDR R1,=0x40000000 ; Resetting the base memory address
```

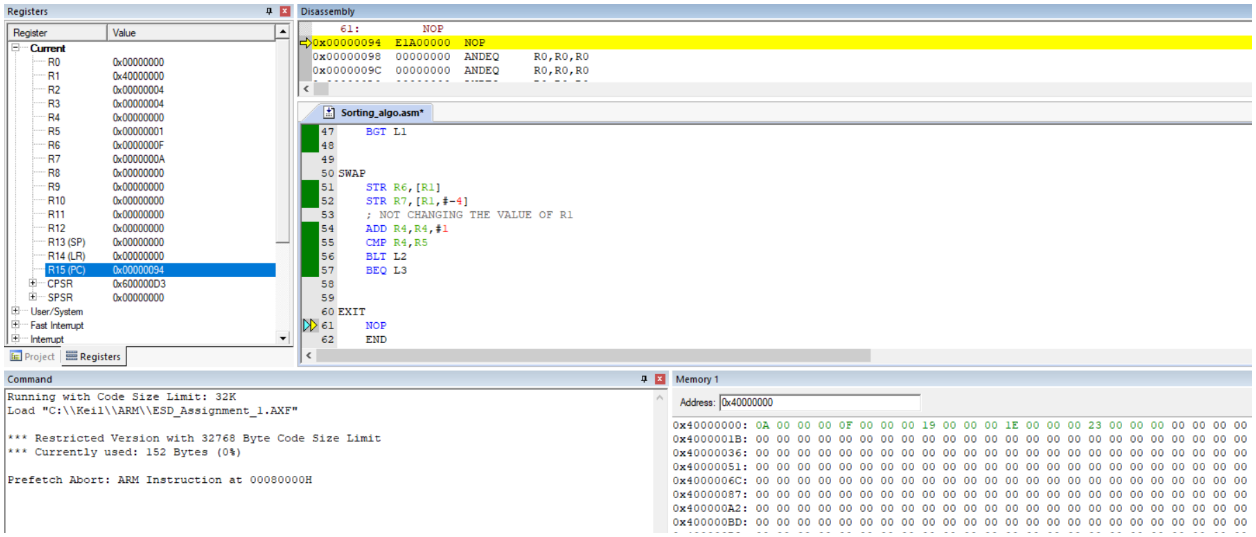
Step-2 : Choose a sorting algorithm and write the assembly code for the algorithm. In this program , Bubble sort has been chosen.



```
; Using the Bubble sort algorithm
L1
    CMP R2,R3 ; if n-1 and i are equal : then end the outer loop
    BEQ EXIT
    SUB R5,R2,R3 ; storing n-i-1 in R5

L2
    LDR R6,[R1],#4
    LDR R7,[R1]
    CMP R6,R7
    BGT SWAP
    ADD R4,R4,#1
    CMP R4,R5
    BLT L2

L3
    ADD R3,R3,#1
    LDR R4,=0x00
    LDR R1,=0x40000000 ; Resetting the base memory address
    CMP R2,R3
    BGT L1

SWAP
    STR R6,[R1]
    STR R7,[R1,#-4]
    ; NOT CHANGING THE VALUE OF R1
    ADD R4,R4,#1
    CMP R4,R5
    BLT L2
    BEQ L3

EXIT
    NOP
    END
```

Step-3: Save the file as an ASM file , add it to the target group. Translate -> Build -> Rebuild. Now run the Debug session to run the code lines one by one. Upon completion the following memory window will appear.

The memory window shows the numbers in the sorted order :
0A<0F<19<1E<23
i.e. 10<15<25<30<35



```
Register        Value
Current
    R0          0x00000000
    R1          0x40000000
    R2          0x00000004
    R3          0x00000004
    R4          0x00000000
    R5          0x00000001
    R6          0x0000000F
    R7          0x0000000A
    R8          0x00000000
    R9          0x00000000
    R10         0x00000000
    R11         0x00000000
    R12         0x00000000
    R13 (SP)    0x00000000
    R14 (LR)    0x00000000
    R15 (PC)    0x00000094
    CPSR        0x600000D3
    SPSR        0x00000000
User/System
Fast Interrupt
Interrupt
```

```
    61:          NOP
0x00000094  E1A00000  NOP
0x00000098  00000000  ANDEQ      R0,R0,R0
0x0000009C  00000000  ANDEQ      R0,R0,R0
```

Sorting_algo.asm*

```
    BGT L1


SWAP
    STR R6,[R1]
    STR R7,[R1,#-4]
    ; NOT CHANGING THE VALUE OF R1
    ADD R4,R4,#1
    CMP R4,R5
    BLT L2
    BEQ L3


EXIT
    NOP
    END
```

```
Running with Code Size Limit: 32K
Load "C:\\Keil\\ARM\\ESD_Assignment_1.AXF"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 152 Bytes (0%)

Prefetch Abort: ARM Instruction at 00080000H
```

Address: 0x40000000

```
0x40000000: 0A 00 00 00 0F 00 00 00 19 00 00 00 1E 00 00 00 23 00 00 00 00 00 00 00
0x4000001B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000036: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000051: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000006C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000087: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000A2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000BD: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Note : The ASM (Assembly language file) has been included in the submission.

To run the code in Keel uVision , the following steps can be used :

- Create a new Project
- Do not add the Startup.s file to the project
- Add the asm file to the target group
- Save it (in case required , in ".asm" format)
- Translate -> Build -> Rebuild
- Start/Stop Debug session