**190905514 SECTION C** **MOHAMMAD TOFIK**

# WEEK 1 LAB1 :

## Solved Exercises:

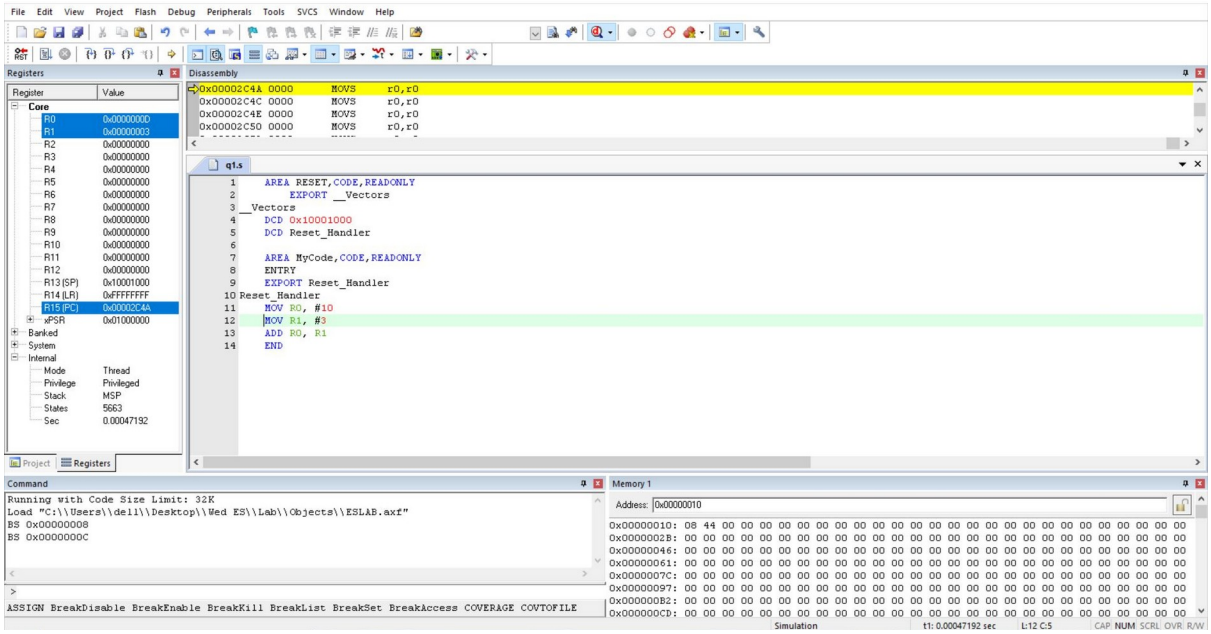
### Example 1

### Program:

```
        AREA RESET,CODE,READONLY
                EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler

        AREA MyCode,CODE,READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        MOV R0, #10
        MOV R1, #3
        ADD R0, R1
        END
```

## Ouptut:




## Example 2

## Program:

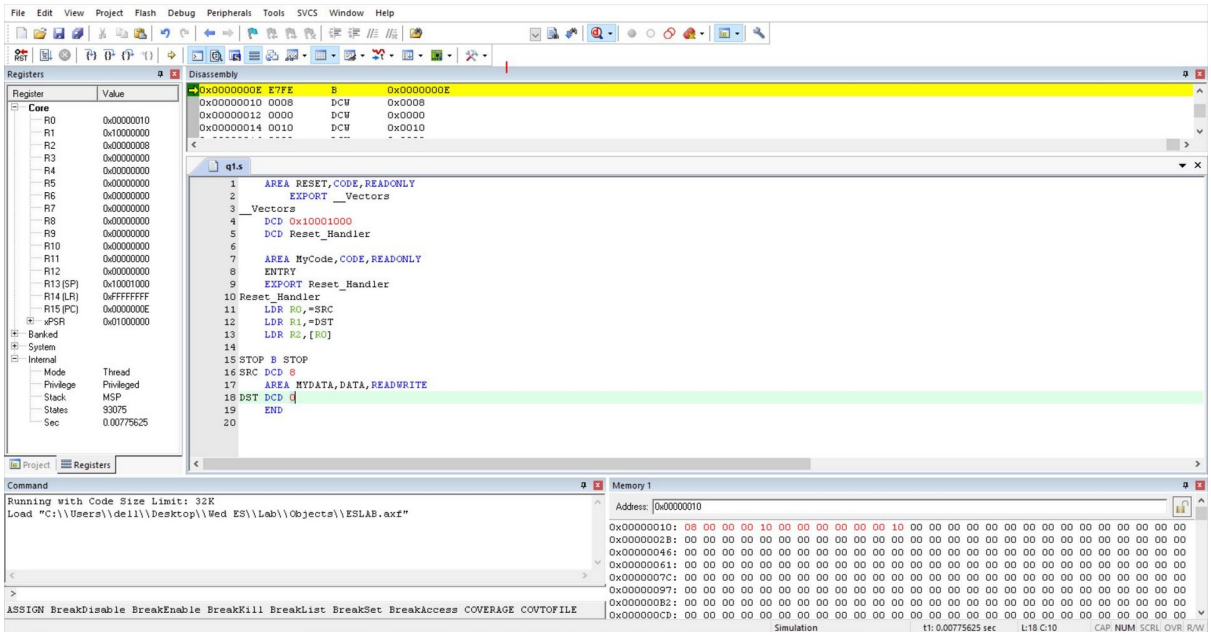
```
        AREA RESET,CODE,READONLY
                EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler

        AREA MyCode,CODE,READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0,=SRC
        LDR R1,=DST
        LDR R2,[R0]

STOP B STOP
SRC DCD 8
        AREA MYDATA,DATA,READWRITE
DST DCD 0
```

END

**Output:**




## Lab Exercises:

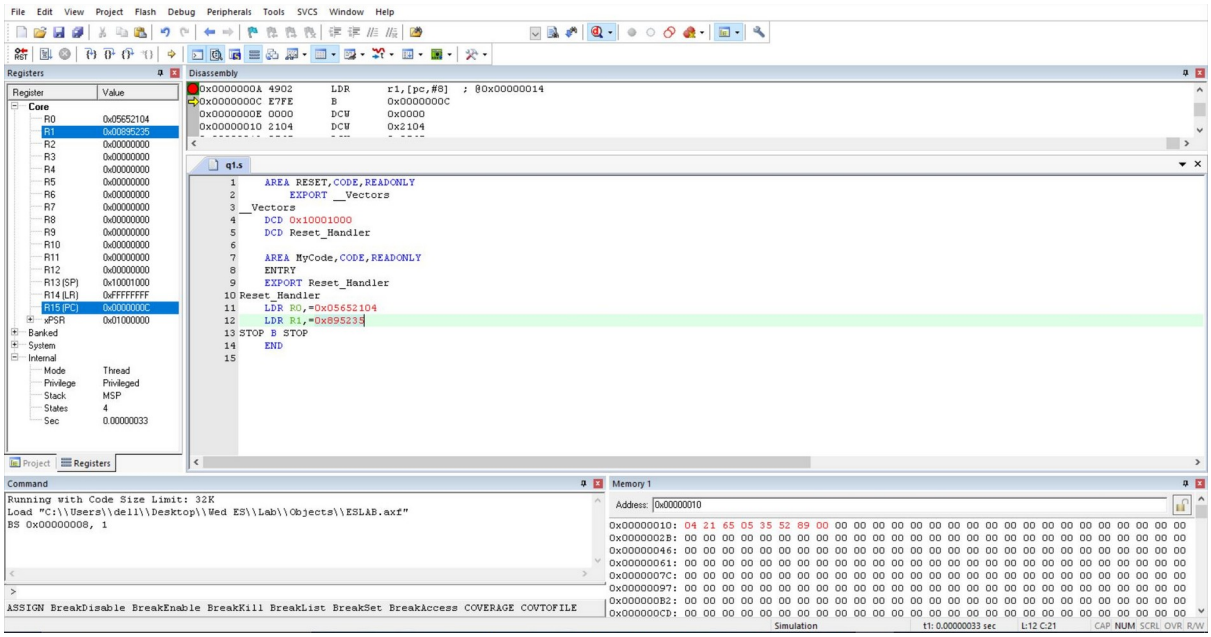
**1.** Write an ARM assembly language program to store data into general purpose registers.

### Program:

```
        AREA RESET,CODE,READONLY
                EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler

        AREA MyCode,CODE,READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0,=0x05652104
        LDR R1,=0x895235
STOP B STOP
        END
```

## Output:




**2.** Write an ARM assembly language program to transfer a 32-bit number from one location in the data memory to another location in the data memory.

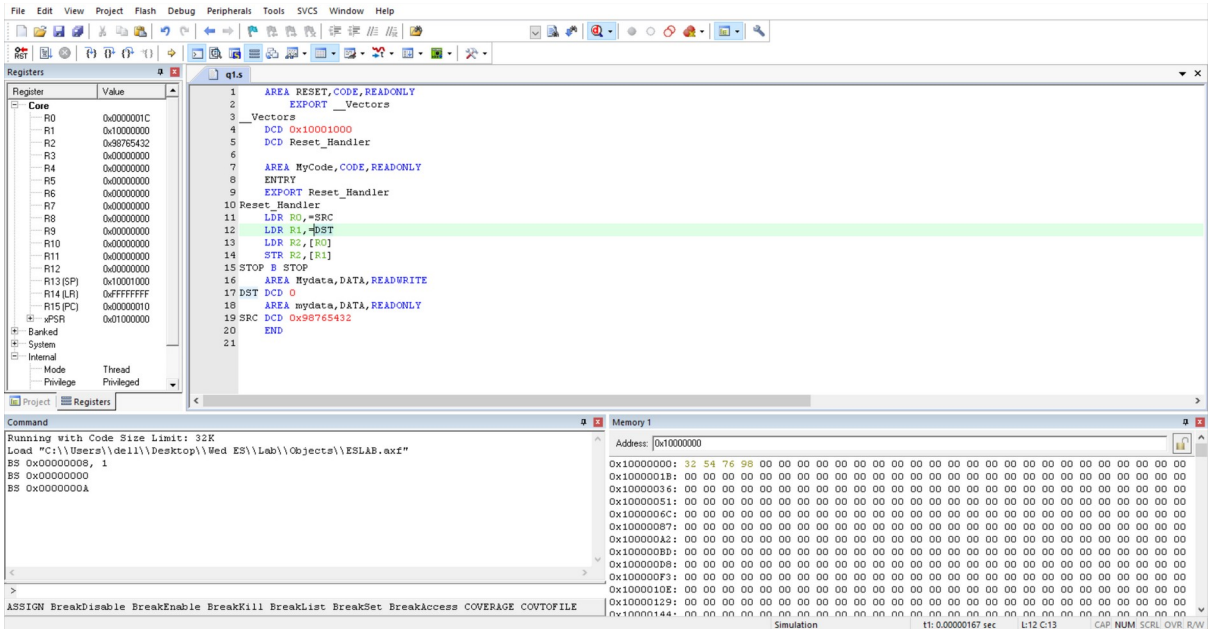
### Program:

```
        AREA RESET,CODE,READONLY
            EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler

        AREA MyCode,CODE,READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0,=SRC
        LDR R1,=DST
        LDR R2,[R0]
        STR R2,[R1]
STOP B STOP
        AREA Mydata,DATA,READWRITE
```

```
DST DCD 0
        AREA mydata,DATA,READONLY
SRC DCD 0x98765432
        END
```

## Output:




**3.** Write an ARM assembly language program to transfer block of ten 32-bit numbers from code memory to data memory when the source and destination blocks are non-overlapping.

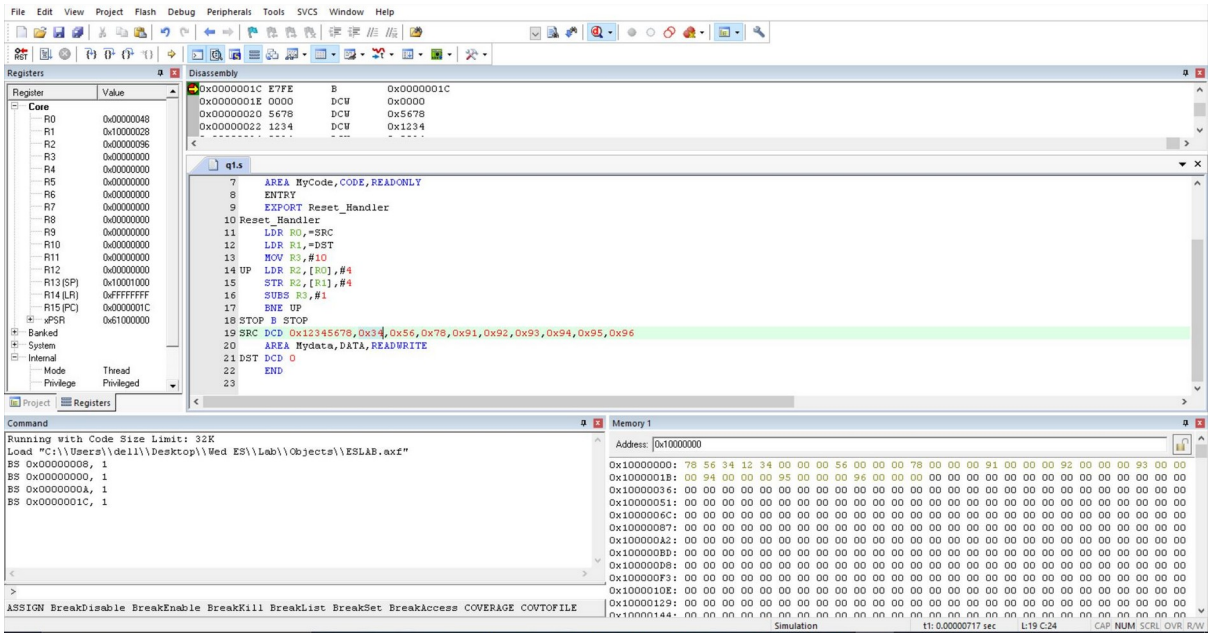
### Program:

```
        AREA RESET,CODE,READONLY
            EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler

        AREA MyCode,CODE,READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0,=SRC
        LDR R1,=DST
        MOV R3,#10
UP      LDR R2,[R0],#4
        STR R2,[R1],#4
        SUBS R3,#1
        BNE UP
STOP B STOP
SRC DCD 0x12345678,0x34,0x56,0x78,0x91,0x92,0x93,0x94,0x95,0x96
        AREA Mydata,DATA,READWRITE
DST DCD 0
```

END

## Output:




**4.** Reverse an array of ten 32-bit numbers in the memory.

### Program:

```
AREA RESET,CODE,READONLY
                EXPORT __Vectors
__Vectors
        DCD 0x10001000
        DCD Reset_Handler


        AREA MyCode,CODE,READONLY
        ENTRY
        EXPORT Reset_Handler
Reset_Handler
        LDR R0,=SRC
        LDR R1,=DST
        ADD R0,R0,#36
        MOV R3,#10
```

```
UP      LDR R2,[R0],#-4
        STR R2,[R1],#4
        SUBS R3,#1
        BNE UP
STOP B STOP
SRC DCD 0x10,0x20,0x30,0x40,0x50,0x60,0x70,0x80,0x90,0x100
        AREA Mydata,DATA,READWRITE
DST DCD 0
        END
```

**Output:**