MICROCONTROLLER AND EMBEDDED SYSTEM SOFTWARE LAB(18CSL48)
PROGRAMS WITH OUTPUT

SAMPLE PROGRAM

AREA SAM,CODE,READONLY

START

       MOV R0,#5

       CMP R0,#0

       MOVLE R0,#0

       MOVGT R0,#1

STOP B STOP

       END

1. PROGRAM TO MULTIPLY TWO 16 BIT BINARY NUMBERS.

```
        AREA MUL,CODE,READONLY

START
     MOV R1,#6400 ;STORE FIRST NUMBER IN R1
     MOV R2,#3200 ; STORE SECOND NUMBER IN R2
     MUL R3,R1,R2  ; MULTIPLICATION
STOP B STOP
     END   ;end of file
```
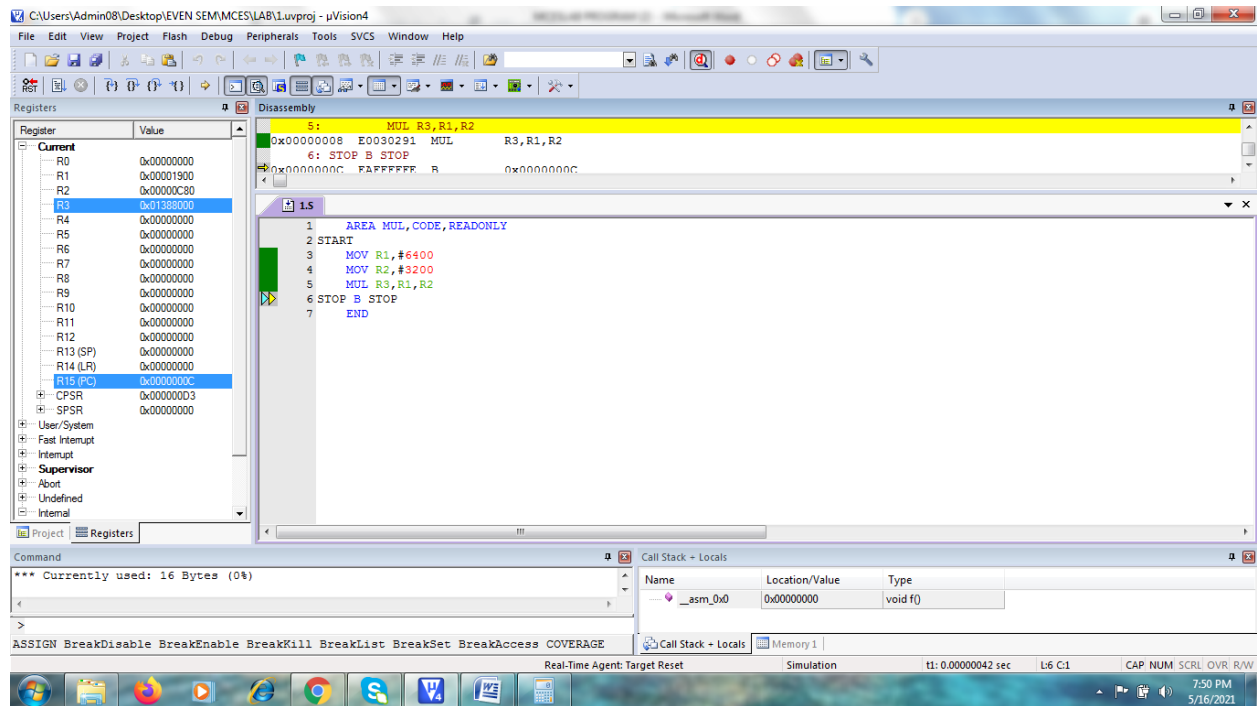
OR

```
        AREA MUL,CODE,READONLY

START

        LDR  R2,=0X23FF    ; LOAD FIRST NUMBER IN R2
        LDR R3,=0X12FF     ; LOAD SECOND NUMBER IN R3

        MUL R4,R3,R2  ;MULTIPLICATION

STOP B STOP

        END
```
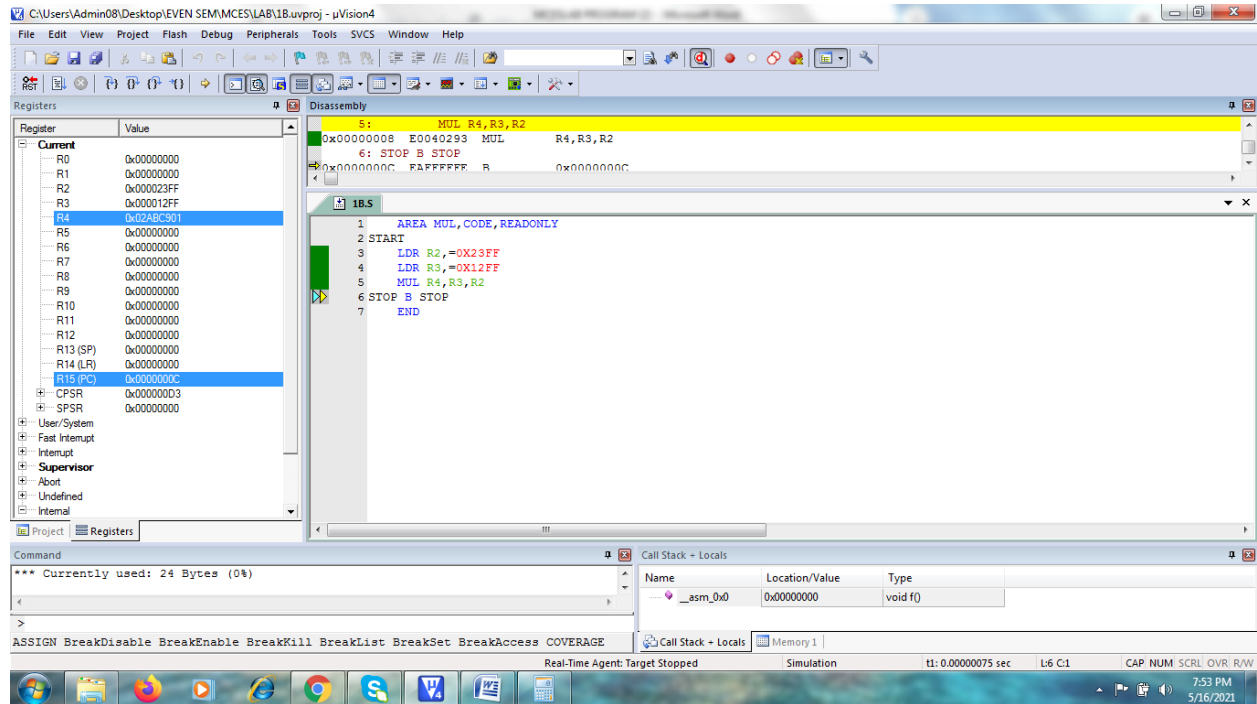
OR

```
        AREA ADD,CODE,READONLY

START

        LDR R0,=X  ; LOAD THE ADDRESS OF X IN TO R0

        LDR R1,[R0]  ; LOAD THE CONTENT OF R0 IN TO R1

        LDR R2,=Y   ; LOAD THE ADDRESS OF Y IN TO R2

        LDR R3,[R2]   ;LOAD THE CONTENT OF R2 IN TO R3

        MUL R4,R3,R1   ;MULTIPLICATION

        LDR R5, =RESULT  ; LOAD THE ADDRESS OF RESULT IN TO R5

        STR R4,[R5]     ; STORE THE R4 CONTENTS IN TO R5 ADDRESS

X DCD 0X4444

Y DCD 0X5555

        AREA SAMPLE,DATA,READWRITE

RESULT DCD 0X00000000

        END
```
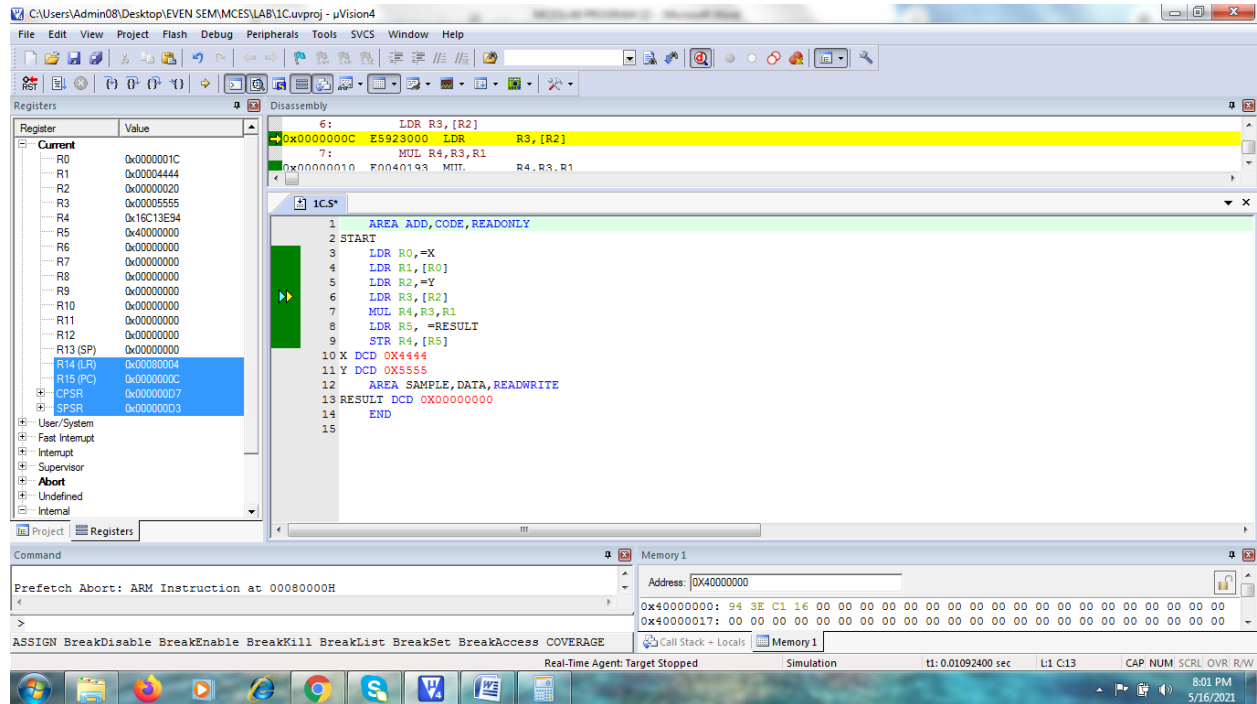
2. WRITE A PROGRAM TO FIND FACTORIAL OF A NUMBER

       AREA FACT,CODE,READONLY

START

       MOV R0,#7

       MOV R1,#1

LOOP

       MUL R2,R1,R0

       MOV R1,R2

       SUBS R0,R0,#1

       BNE LOOP ;BRANCH IF IT'S NOT EQUAL TO ZERO GO BACK TO LOOP

STOP B STOP

END
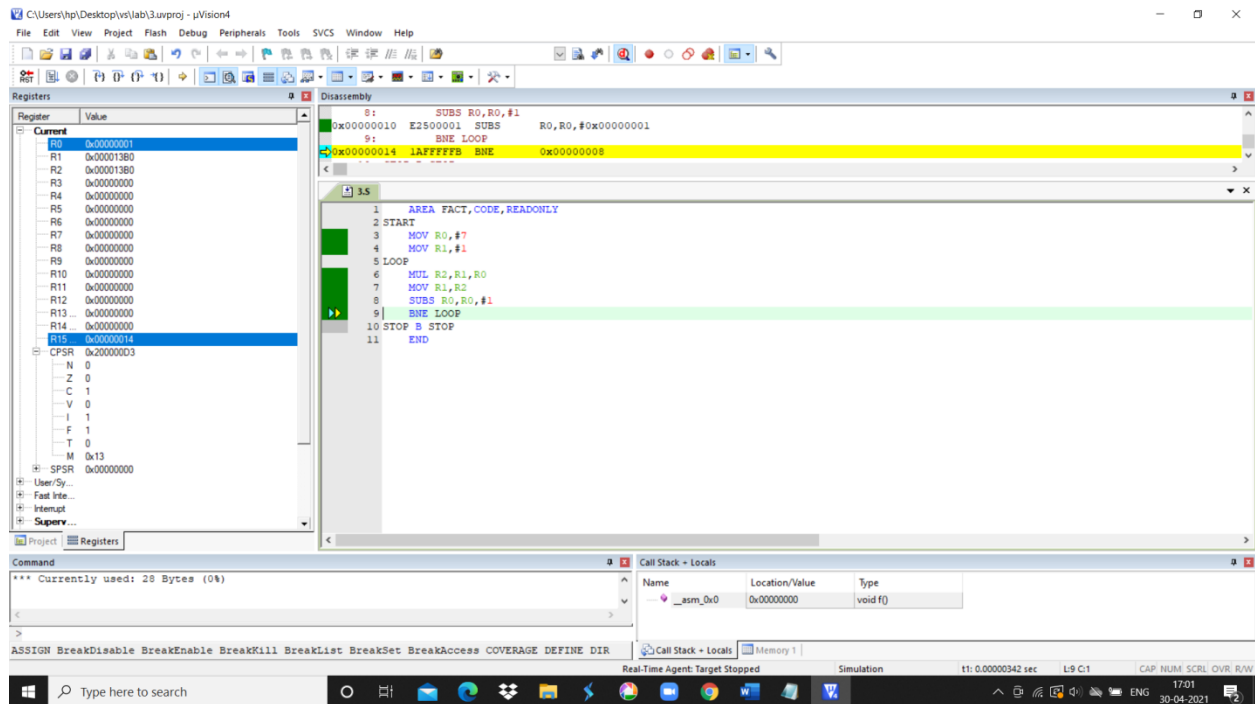
3. WRITE A PROGRAM TO FIND THE SUM OF FIRST 10 INTEGER NUMBERS

AREA SUM,CODE,READONLY

START

MOV R0,#10

MOV R1,#00

LOOP

ADD R1,R1,R0

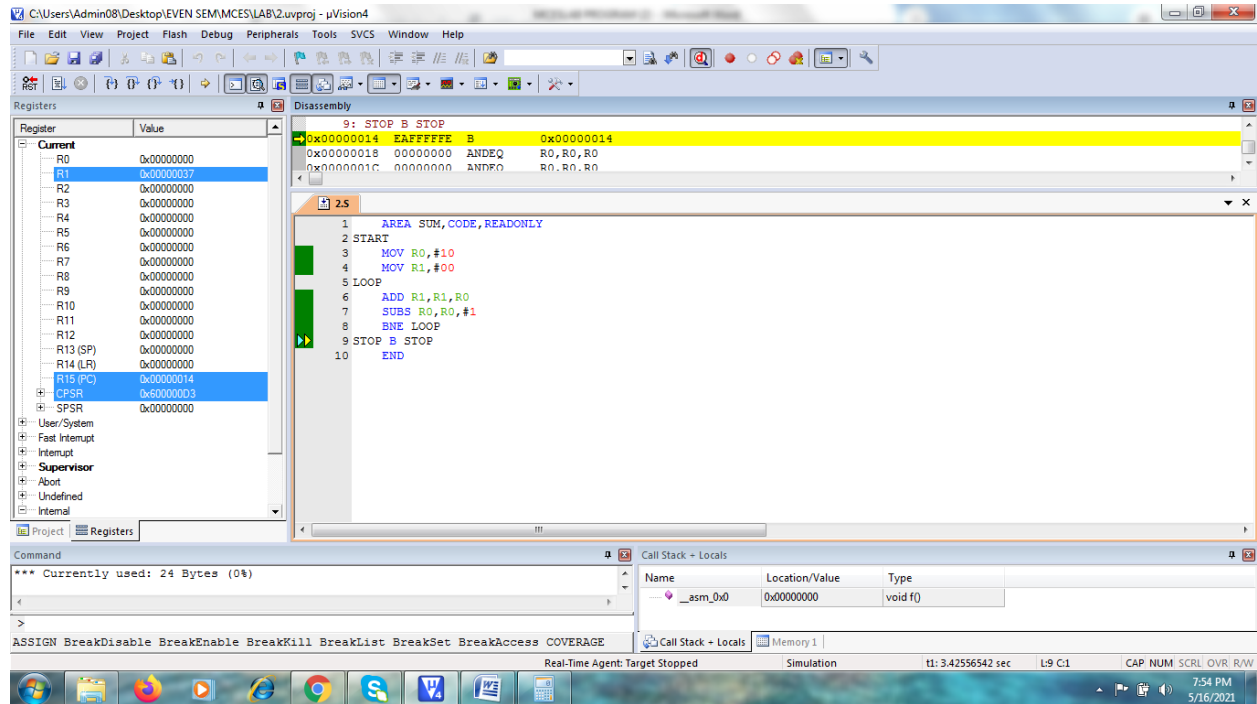SUBS R0,R0,#1

BNE LOOP

STOP B STOP

END

4. PROGRAM TO ADD AN ARRAY OF 16 BIT NUMBERS AND STORE THE 32 BIT RESULT IN INTERNAL RAM.

AREA ARRAY,CODE,READONLY

START

       MOV R0,#0

       LDR R1,=TABLE

       MOV R2,#6

LOOP

       LDRH R3,[R1],#2; first data is loaded in to R3 , then address is updated.

       ADD R0,R0,R3

       SUBS R2,R2,#1

       BNE LOOP

       LDR R1,=0X40000080

       STR R0,[R1]

STOP B STOP

TABLE DCW 0X1111,0X2222,0X3333,0X4444,0X5555,0X6666

END

OR

AREA ARRAY,CODE,READONLY

START

MOV R0,#0

LDR R1,=TABLE

MOV R2,#6

LOOP

LDRH R3,[R1],#2 ; first data is loaded in to R3 , then address is updated.

ADD R0,R0,R3

SUBS R2,R2,#1

BNE LOOP

LDR R1,=RESULT

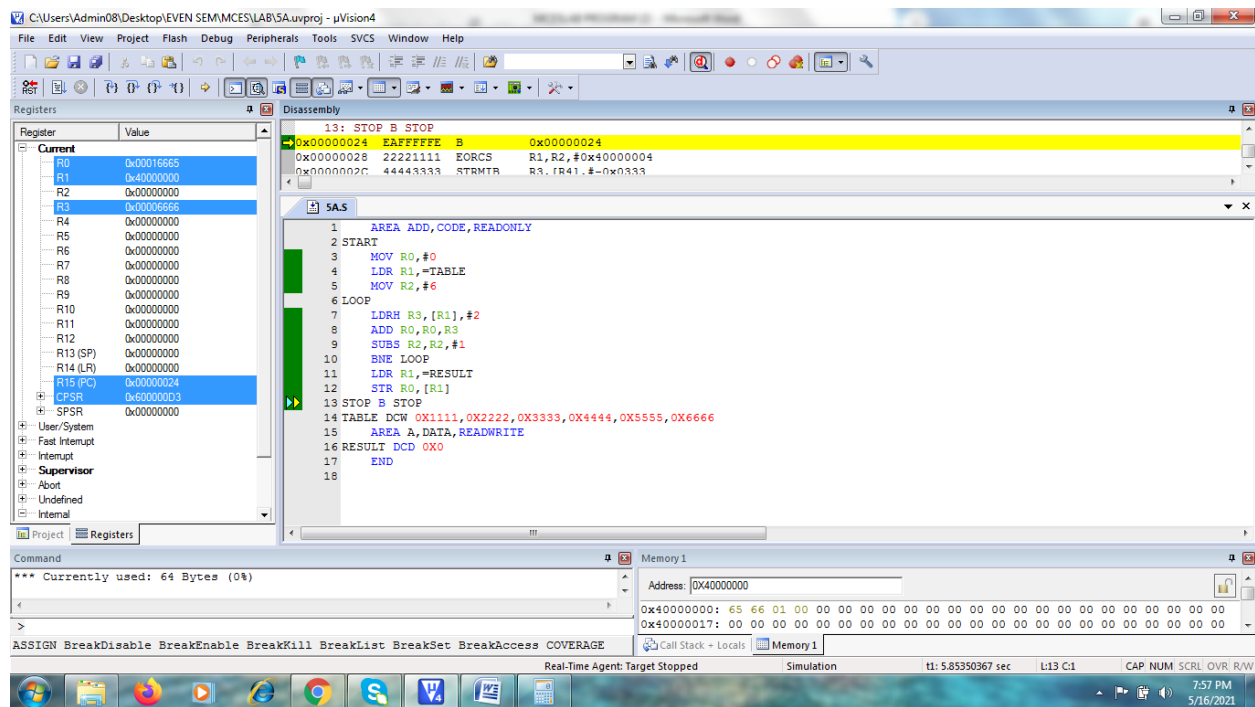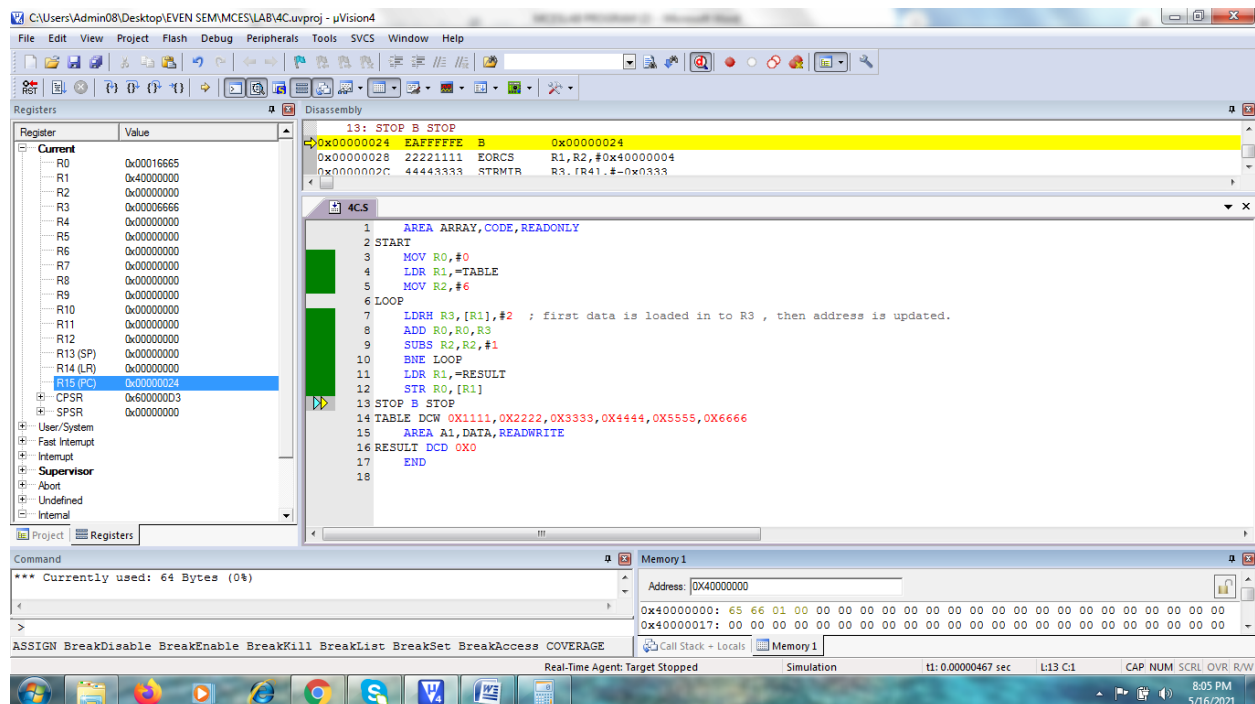STR R0,[R1]

STOP B STOP

TABLE DCW 0X1111,0X2222,0X3333,0X4444,0X5555,0X6666

    AREA A1,DATA,READWRITE

RESULT DCD 0X0

    END

5.PROGRAM TO FIND THE SQUARE OF A NUMBER(1 TO 10) USING LOOK-UP TABLE

    AREA SQUARE,CODE,READONLY

START

    MOV R2,#00

    MOV R1,#7 ;MOV 7 IN TO R1

    LDR R0,=LOOKUP ; LOAD THE ADDRESS OF LOOKUP IN TO R0

    LDRB R2,[R0,R1] ; R0 + R1(7) GO TO THAT LOCATION GET THE SQUARE AND LOAD IN TO R2
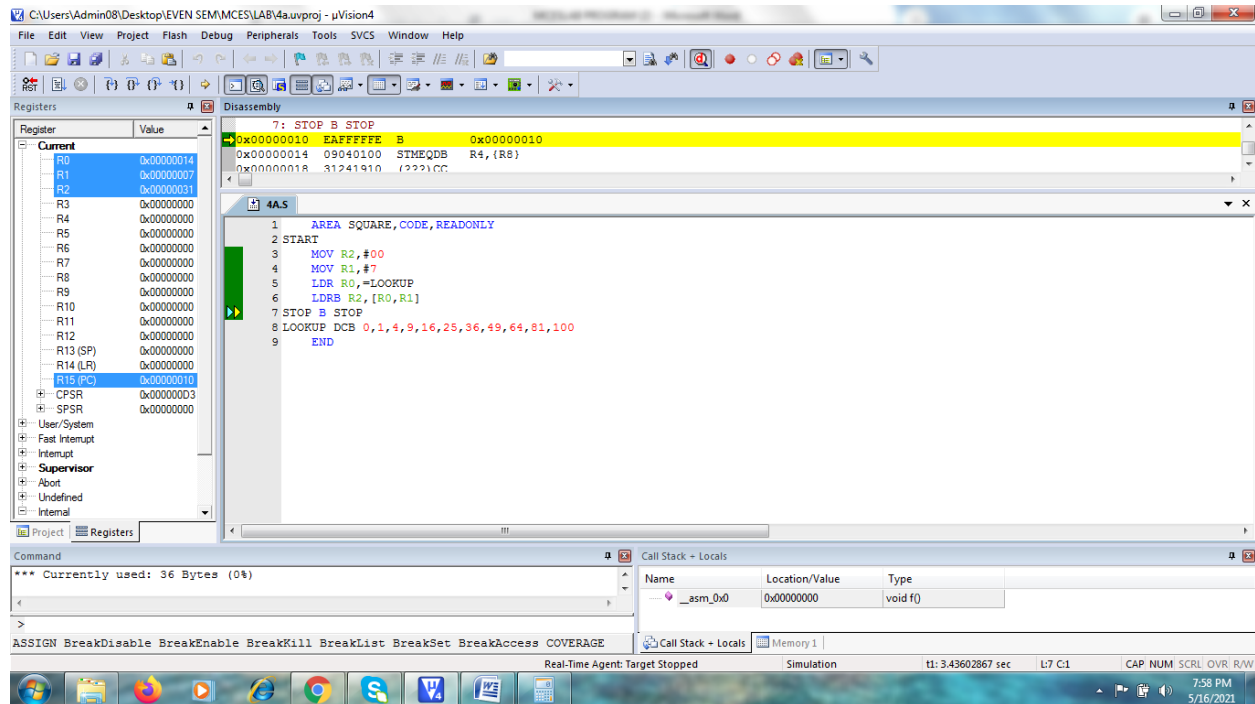
STOP B STOP

LOOKUP DCB 0,1,4,9,16,25,32,49,64,81,100

END

6. PROGRAM TO FIND THE LARGET/SMALLEST NUMBERS IN AN ARRAY OF 32- BIT NUMBERS.

LARGET NUMBERS

AREA LARG,CODE,READONLY

START

MOV R5,#6 ;INITIALISE COUNTER TO 6

LDR R1,=VALUE ;LAODS THE ADDRESS OF FIRST VALUE

LDR R2,[R1],#4 ;LOAD THE R1 CONTENT TO R2 AND INCREAMENT THE R1 SO THAT IT CAN POINT TO NEXT ADDRESS

LOOP

LDR R4,[R1],#4 ; LOAD THE R1 CONTENT TO R4 AND INCREAMENT THE R1 SO THAT IT CAN POINT TO NEXT ADDRESS

CMP R2,R4 ;COMPARE R2 AND R4 CONTENTS

BHI LOOP1 ;BRANCH HIGHER(IF THE FIRST NUMBER IS> THEN GO TO LOOP1)

MOV R2,R4 ;IF THE FIRST NUMBER IS< THEN MOV CONTENT R4 TO R2

LOOP1

SUBS R5,R5,#1 ; DECREMENT COUNTER

CMP R5,#0 ;COMPARE COUNTER TO 0

BNE LOOP ;LOOP BACK TILL ARRAY ENDS

LDR R4,=RESULT ;LOADS THE ADDRESS OF RESULT

STR R2,[R4] ;STORES THE RESULT IN R2

STOP B STOP

VALUE

DCD 0X44444444

DCD 0X22222222

DCD 0X11111111

DCD 0X33333333

DCD 0XAAAAAAAA

DCD 0X88888888

DCD 0X99999999

AREA DATA2,DATA,READWRITE

RESULT DCD 0X0

END

SMALLEST NUMBERS

AREA LARG,CODE,READONLY

START

        MOV R5,#6

        LDR R1,=VALUE

        LDR R2,[R1],#4

LOOP

        LDR R4,[R1],#4

        CMP R2,R4

        BLS LOOP1 ;BRANCH LESS(BLO)

        MOV R2,R4

LOOP1

        SUBS R5,R5,#1

        CMP R5,#0

        BNE LOOP

```
        LDR R4,=RESULT

        STR R2,[R4]

STOP B STOP

VALUE

        DCD 0X44444444

        DCD 0X22222222

        DCD 0X11111111

        DCD 0X33333333

        DCD 0XAAAAAAAA

        DCD 0X88888888

        DCD 0X99999999

        AREA DATA2,DATA,READWRITE

RESULT DCD 0X0

        END
```
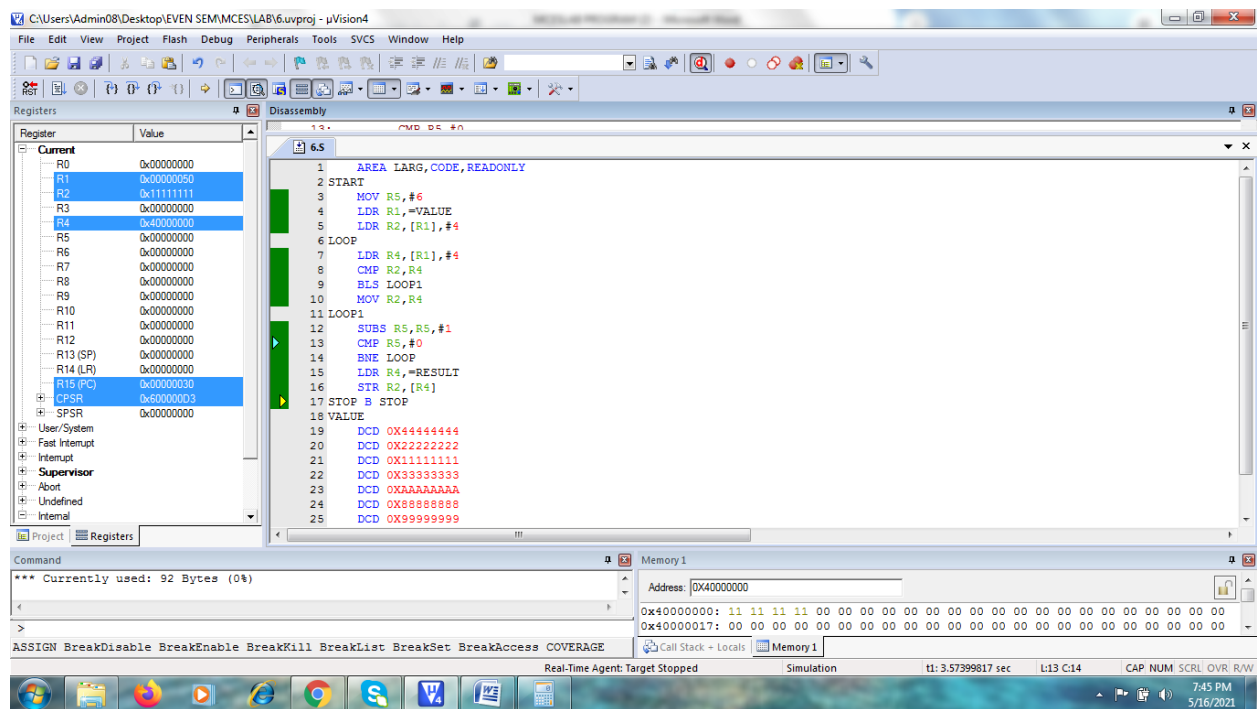
7. PROGRAM TO ARRANGE A SERIES OF 32 BIT NUMBERS IN ASCENDING/DECENDING ORDER

DECENDING ORDER

START

      MOV R1,#5

OUTLOOP

      MOV R3,R1

      LDR R0,=0X40000000

INLOOP

      LDR R2,[R0],#4

      LDR R4,[R0]

      CMP R4,R2

      BLS SKIP ;BHI/BGT

LOOP2

      STR R2,[R0],#-4
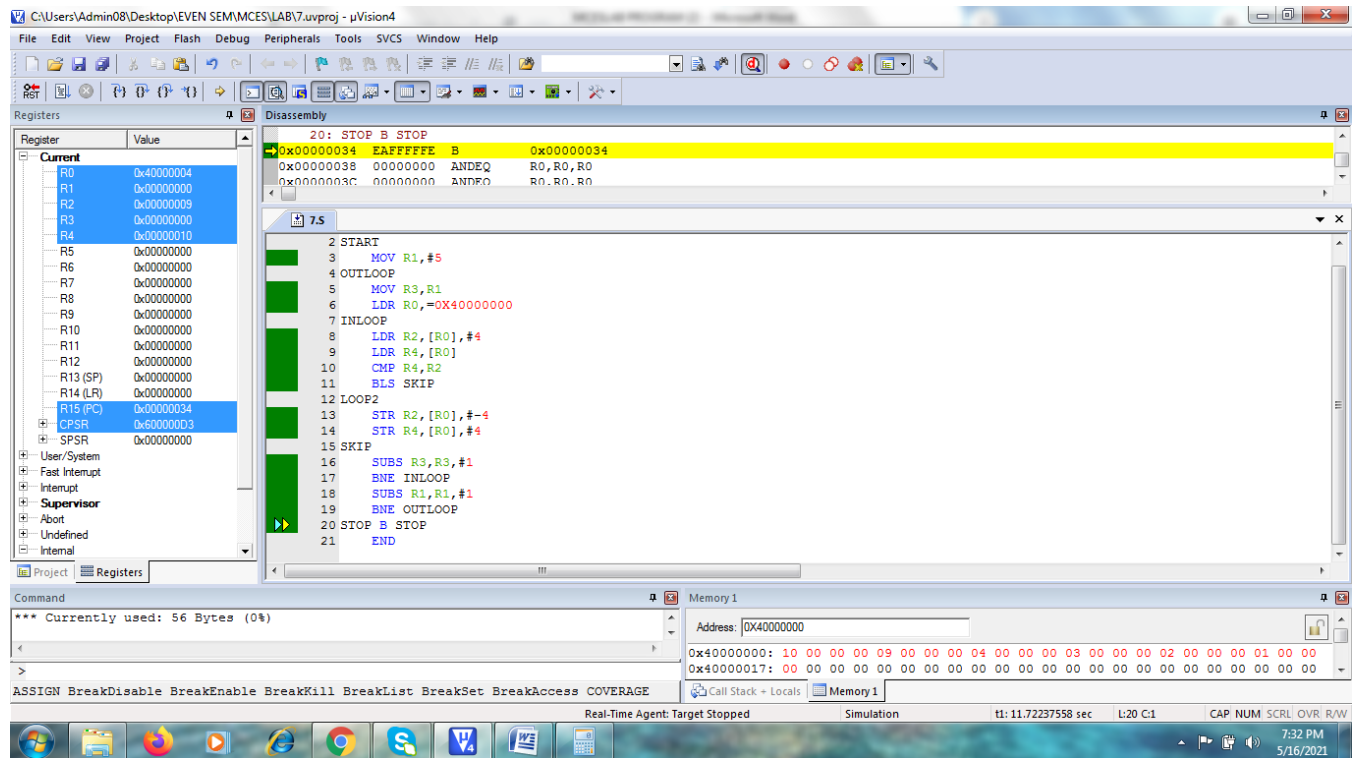
      STR R4,[R0],#4

SKIP

      SUBS R3,R3,#1

      BNE INLOOP

      SUBS R1,R1,#1

      BNE OUTLOOP

STOP B STOP

      END

ASCENDING ORDER

AREA SORT,CODE,READONLY

START

MOV R1,#5

OUTLOOP

MOV R3,R1

LDR R0,=0X40000000

INLOOP

LDR R2,[R0],#4

LDR R4,[R0]

CMP R4,R2

BHI SKIP

LOOP2

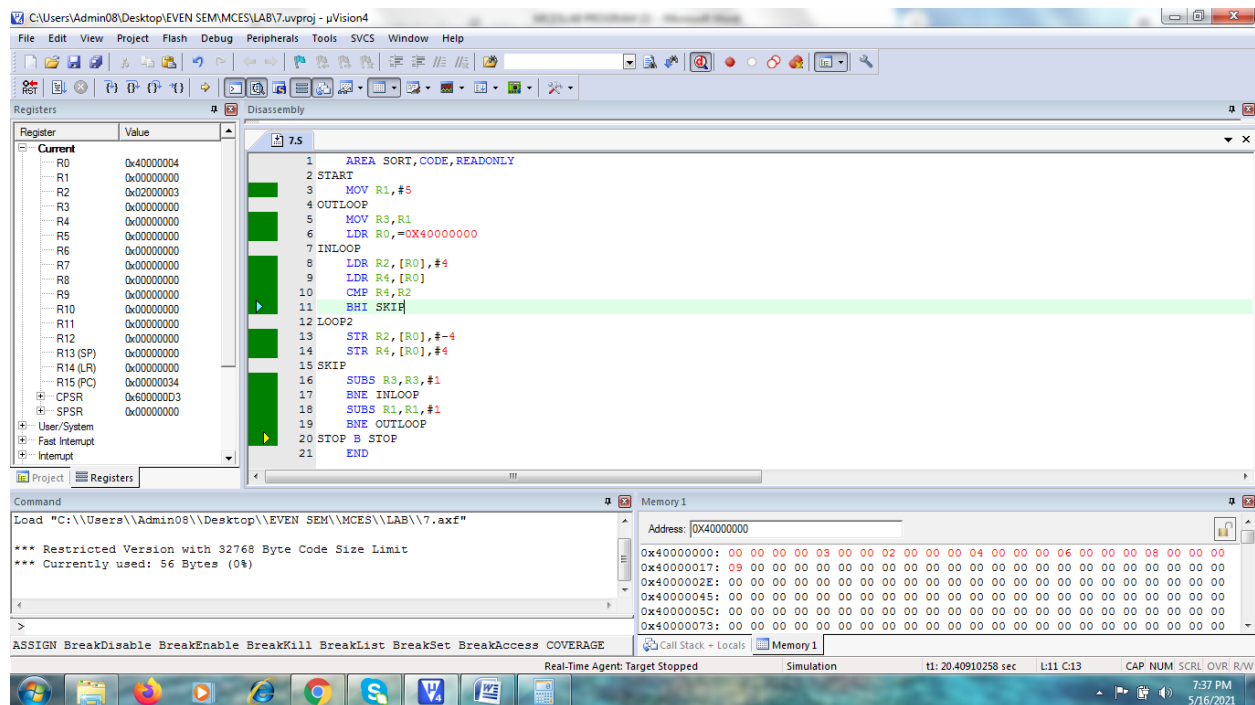STR R2,[R0],#-4

STR R4,[R0],#4

SKIP

SUBS R3,R3,#1

BNE INLOOP

SUBS R1,R1,#1

BNE OUTLOOP

STOP B STOP

END

8. PROGRAM TO COUNT THE NUMBER OF ONCES AND ZEROS IN TWO CONSECUTIVE MEMORY LOCATIONS

AREA COUNT,CODE,READONLY

START

LDR R0,=0X40000050

LDRH R1,[R0]

MOV R2,#16

LOOP

MOVS R1,R1,LSR #1

ADDCS R3,#1

ADDCC R4,#1

SUBS R2,#1

BNE LOOP

STOP B STOP

END