Faculty of Engineering & Technology
Electrical & Computer Engineering Department

Computer Organization And Microprocessor

Assembly Assignment

Prepared by:

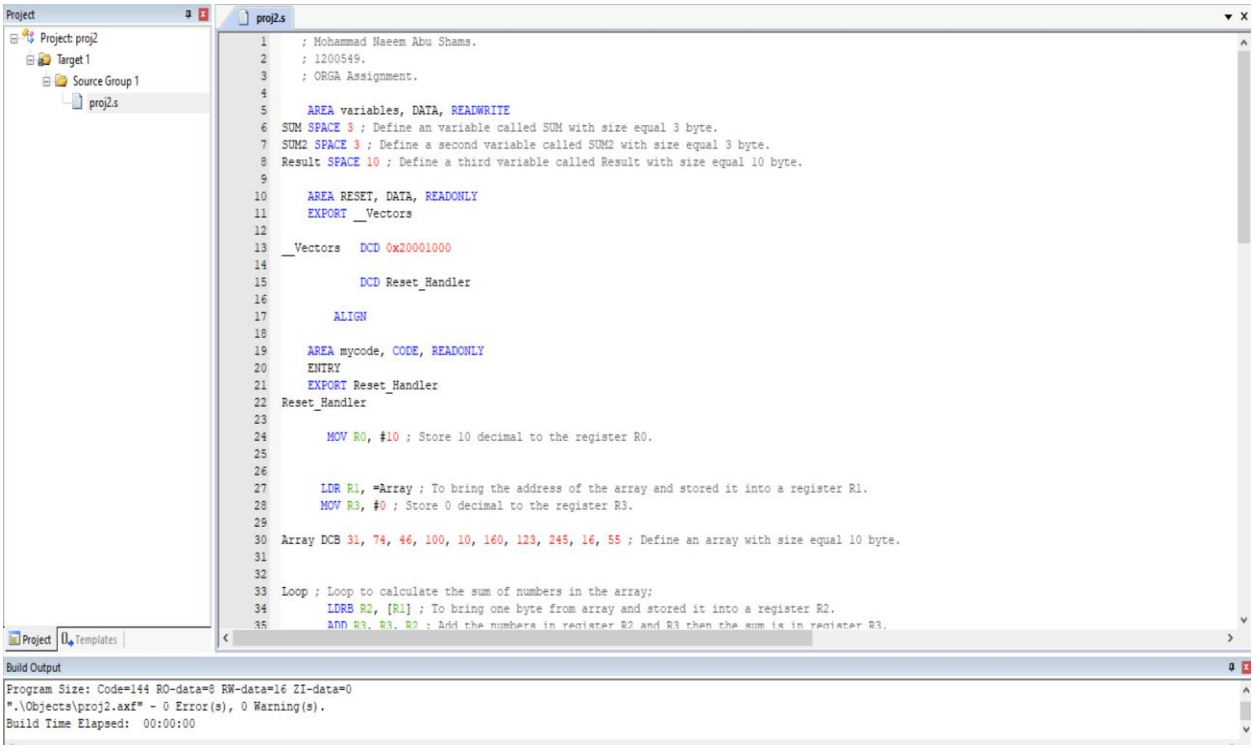
Mohammad Abu Shams 1200549

Instructor: Dr. Ahmad Afaneh

Section: 3

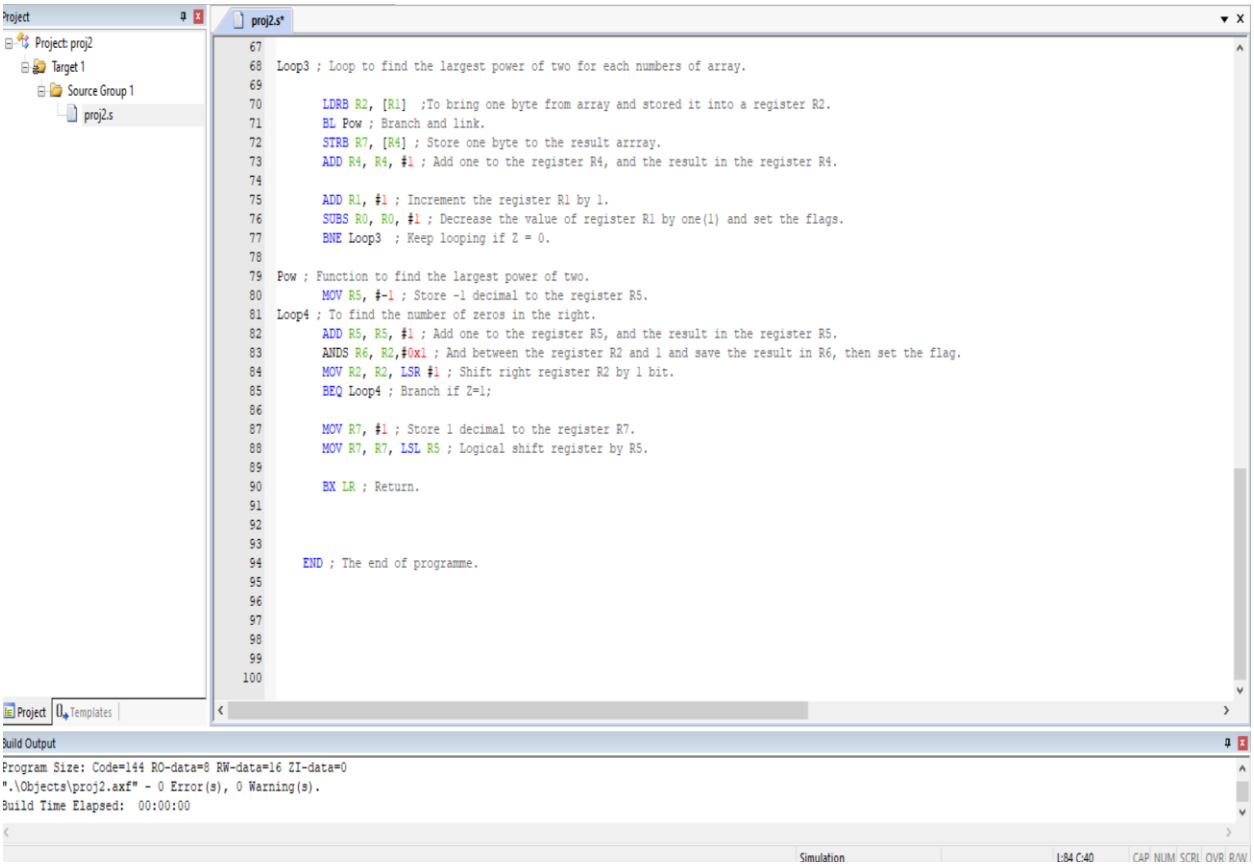
Date: 22-6-2022

The Code:



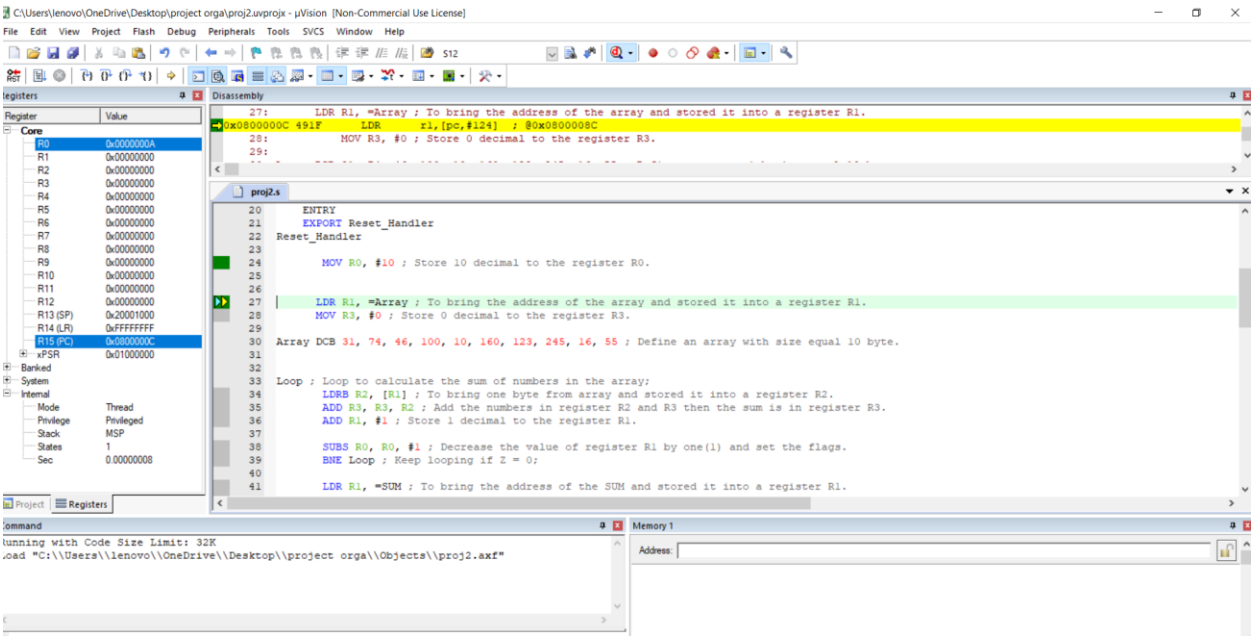
```
; Mohammad Naeem Abu Shams.
; 1200549.
; ORGA Assignment.

    AREA variables, DATA, READWRITE
SUM SPACE 3 ; Define an variable called SUM with size equal 3 byte.
SUM2 SPACE 3 ; Define a second variable called SUM2 with size equal 3 byte.
Result SPACE 10 ; Define a third variable called Result with size equal 10 byte.

    AREA RESET, DATA, READONLY
    EXPORT __Vectors

__Vectors   DCD 0x20001000

            DCD Reset_Handler

        ALIGN

    AREA mycode, CODE, READONLY
    ENTRY
    EXPORT Reset_Handler
Reset_Handler

       MOV R0, #10 ; Store 10 decimal to the register R0.


      LDR R1, =Array ; To bring the address of the array and stored it into a register R1.
      MOV R3, #0 ; Store 0 decimal to the register R3.

Array DCB 31, 74, 46, 100, 10, 160, 123, 245, 16, 55 ; Define an array with size equal 10 byte.


Loop ; Loop to calculate the sum of numbers in the array;
       LDRB R2, [R1] ; To bring one byte from array and stored it into a register R2.
       ADD R3, R3, R2 ; Add the numbers in register R2 and R3 then the sum is in register R3.
       ADD R1, #1 ; Store 1 decimal to the register R1.

       SUBS R0, R0, #1 ; Decrease the value of register R1 by one(1) and set the flags.
       BNE Loop ; Keep looping if Z = 0;

       LDR R1, =SUM ; To bring the address of the SUM and stored it into a register R1.
       STR R3, [R1] ; Stroe the value of R3 to SUM.

       LDR R1, =Array ; To bring the address of the array and stored it into a register R1.
         MOV R0, #10 ; Store 10 decimal to the register R0.
       MOV R4, #0 ;Store 0 decimal to the register R4.

loop2 ; Loop to find the sum of even numbers in the array.

       LDRB R2, [R1] ; To bring one byte from array and stored it into a register R2.

       MOVS R3, R2, LSL #31 ; Shift left register R2 by 31 bit, then set the flags. If Z=0, then the number is odd, else if the Z=1, then the numb
       BNE Skip; If Z=0, branch ( Skip the add ).
       ADD R4, R4, R2; Add the number of register R2 to the register R4, and the result in the register R4.
Skip
       ADD R1, #1 ; Increment the register R1 by 1.
       SUBS R0, R0, #1 ; Decrease the value of register R1 by one(1) and set the flags.

       BNE loop2 ; Keep looping if Z = 0.

       LDR R1, =SUM2 ; To bring the address of the SUM2 and stored it into a register R1.
       STR R4, [R1] ; Stroe the value of R4 to SUM2.

       LDR R1, =Array; To bring the address of the array and stored it into a register R1.
       LDR R4, =Result ; To bring the address of the Result and stored it into a register R4.
       MOV R0, #10 ; Store 10 decimal to the register R0.

Loop3 ; Loop to find the largest power of two for each numbers of array.
```

```
Build Output
Program Size: Code=144 RO-data=8 RW-data=16 ZI-data=0
".\Objects\proj2.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

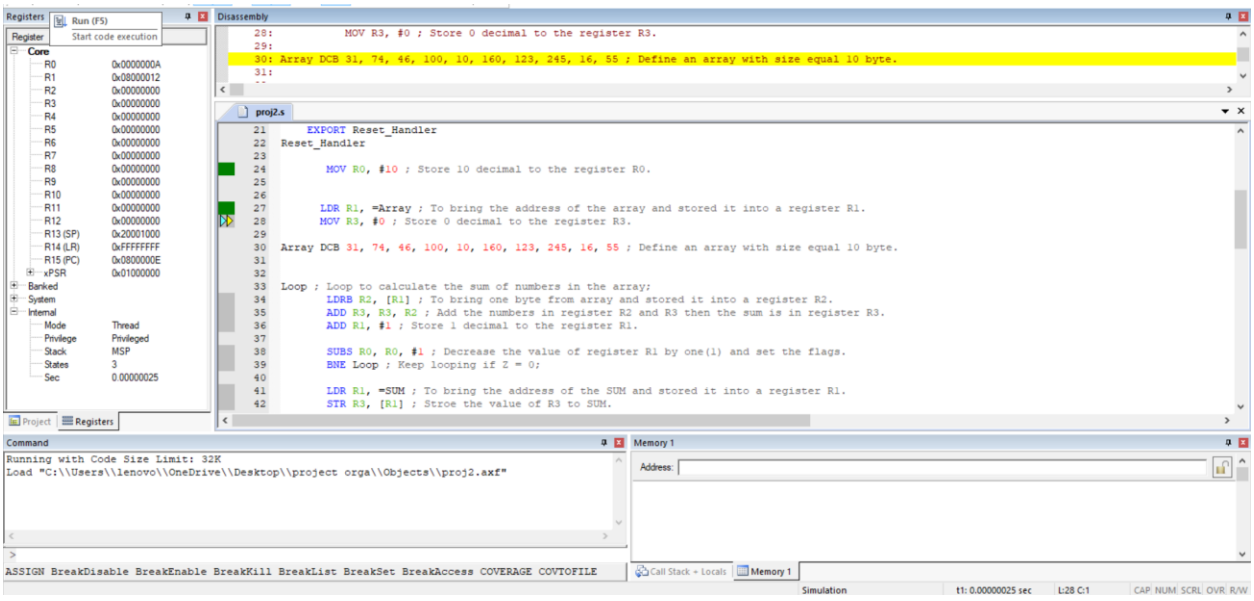

Project
Project: proj2
Target 1
Source Group 1
proj2.s
proj2.s*
Loop3 ; Loop to find the largest power of two for each numbers of array.
LDRB R2, [R1] ;To bring one byte from array and stored it into a register R2.
BL Pow ; Branch and link.
STRB R7, [R4] ; Store one byte to the result arrray.
ADD R4, R4, #1 ; Add one to the register R4, and the result in the register R4.
ADD R1, #1 ; Increment the register R1 by 1.
SUBS R0, R0, #1 ; Decrease the value of register R1 by one(1) and set the flags.
BNE Loop3 ; Keep looping if Z = 0.
Pow ; Function to find the largest power of two.
MOV R5, #-1 ; Store -1 decimal to the register R5.
Loop4 ; To find the number of zeros in the right.
ADD R5, R5, #1 ; Add one to the register R5, and the result in the register R5.
ANDS R6, R2,#0x1 ; And between the register R2 and 1 and save the result in R6, then set the flag.
MOV R2, R2, LSR #1 ; Shift right register R2 by 1 bit.
BEQ Loop4 ; Branch if Z=1;
MOV R7, #1 ; Store 1 decimal to the register R7.
MOV R7, R7, LSL R5 ; Logical shift register by R5.
BX LR ; Return.
END ; The end of programme.
Project
Templates
Build Output
Program Size: Code=144 RO-data=8 RW-data=16 ZI-data=0
".\Objects\proj2.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:00
Simulation
L:84 C:40
CAP NUM SCRL OVR R/W

a) Declare an array of at least 10 8-bit unsigned integer numbers in the memory with initial values.

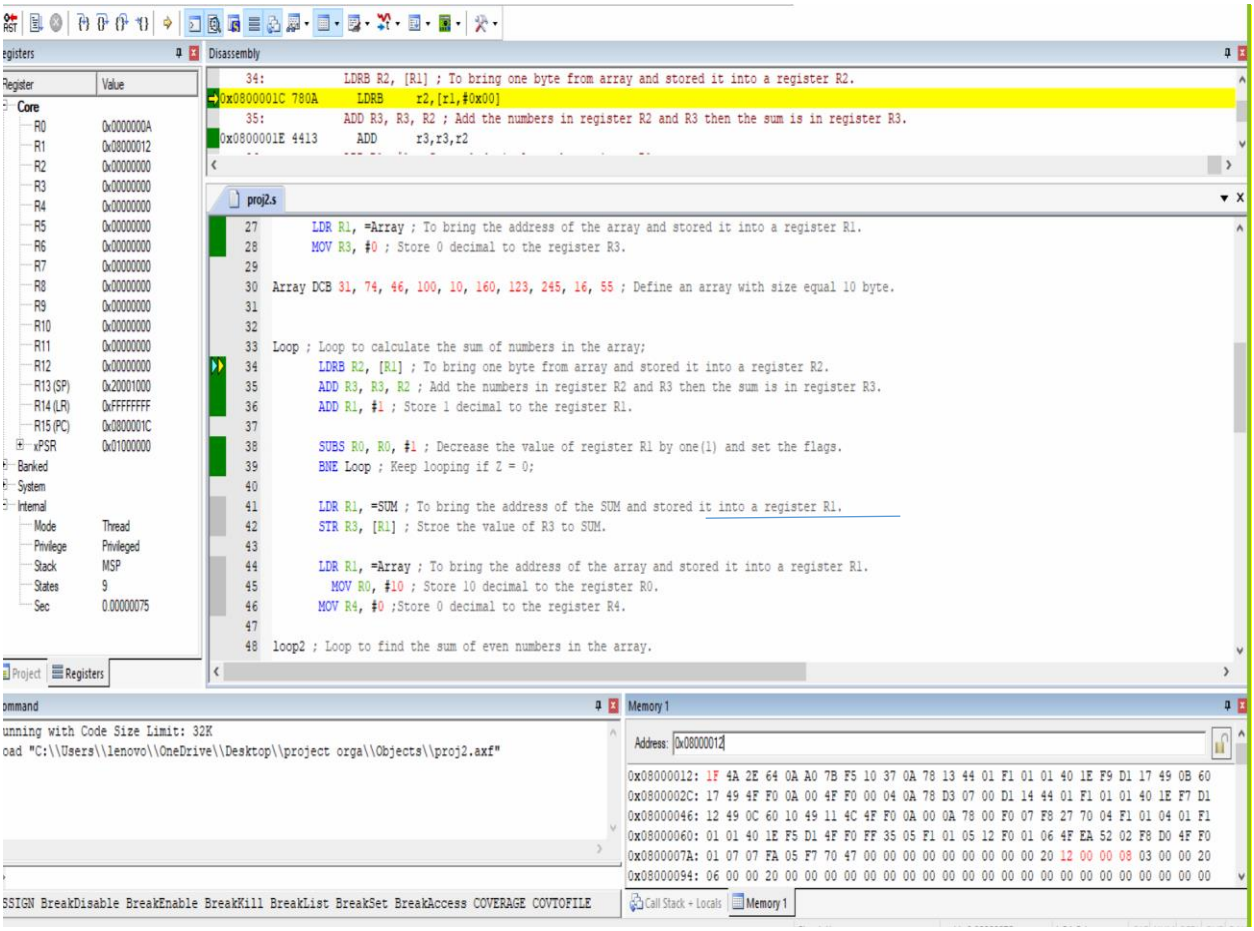
31, 74, 46, 100, 10, 160, 123, 245, 16, 55



Store 10 decimal to the register R0.



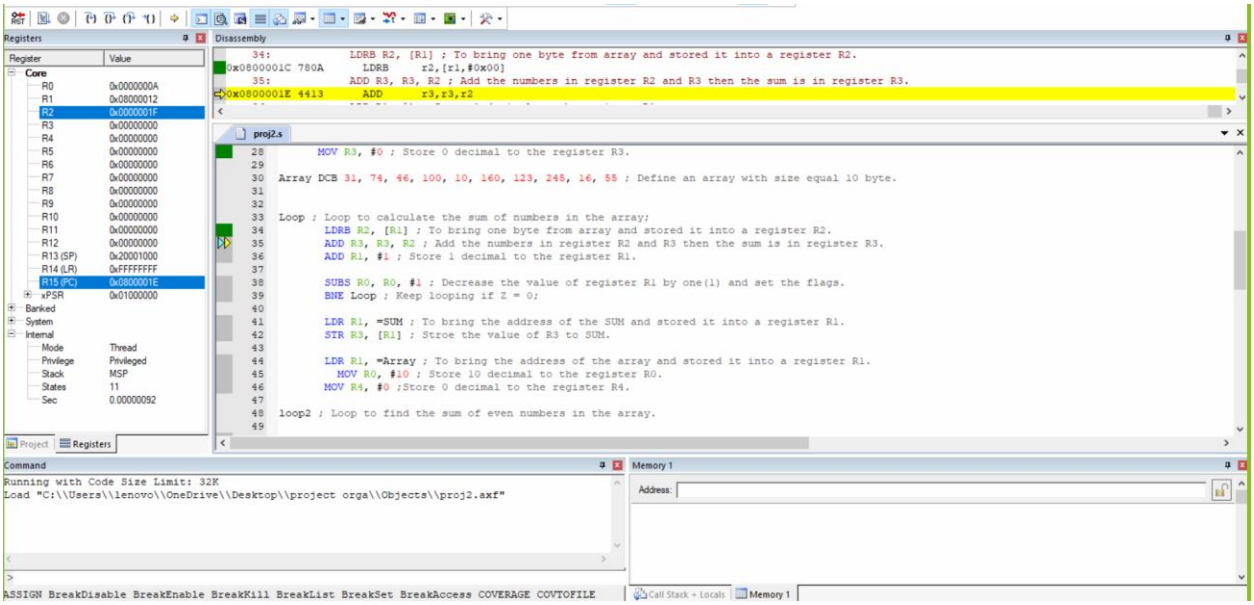
Store the address of the array into a register R1.



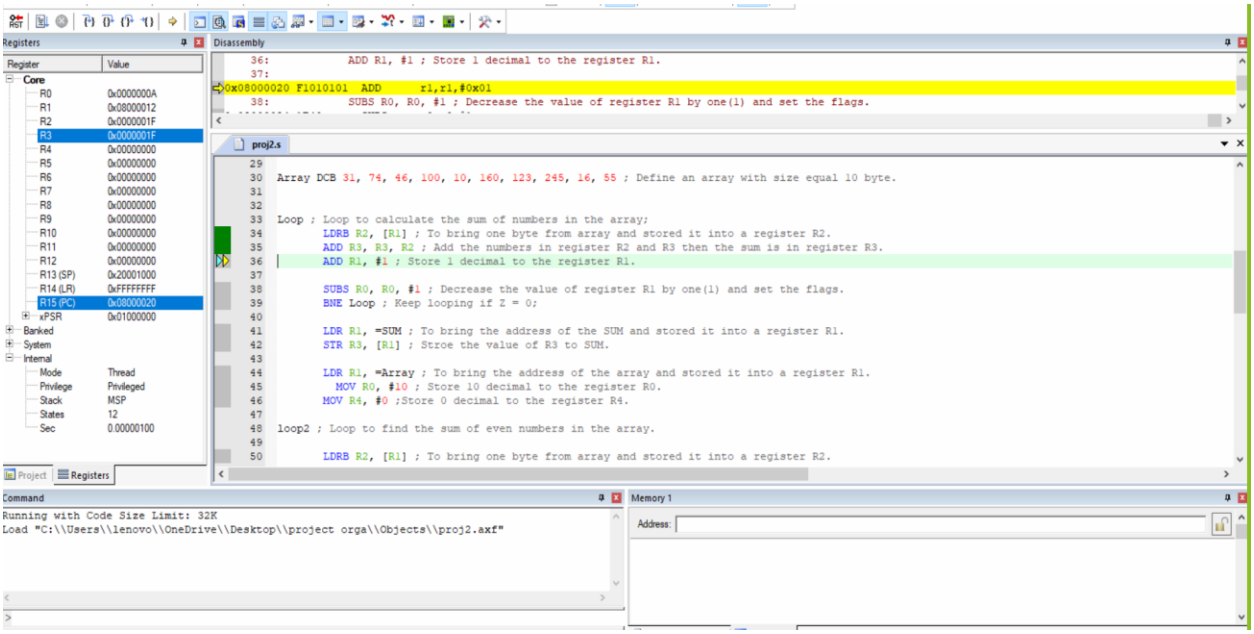
The first 10 number in this address from left to right in the first line which underlined with blue color it's the array values.

The first part is done

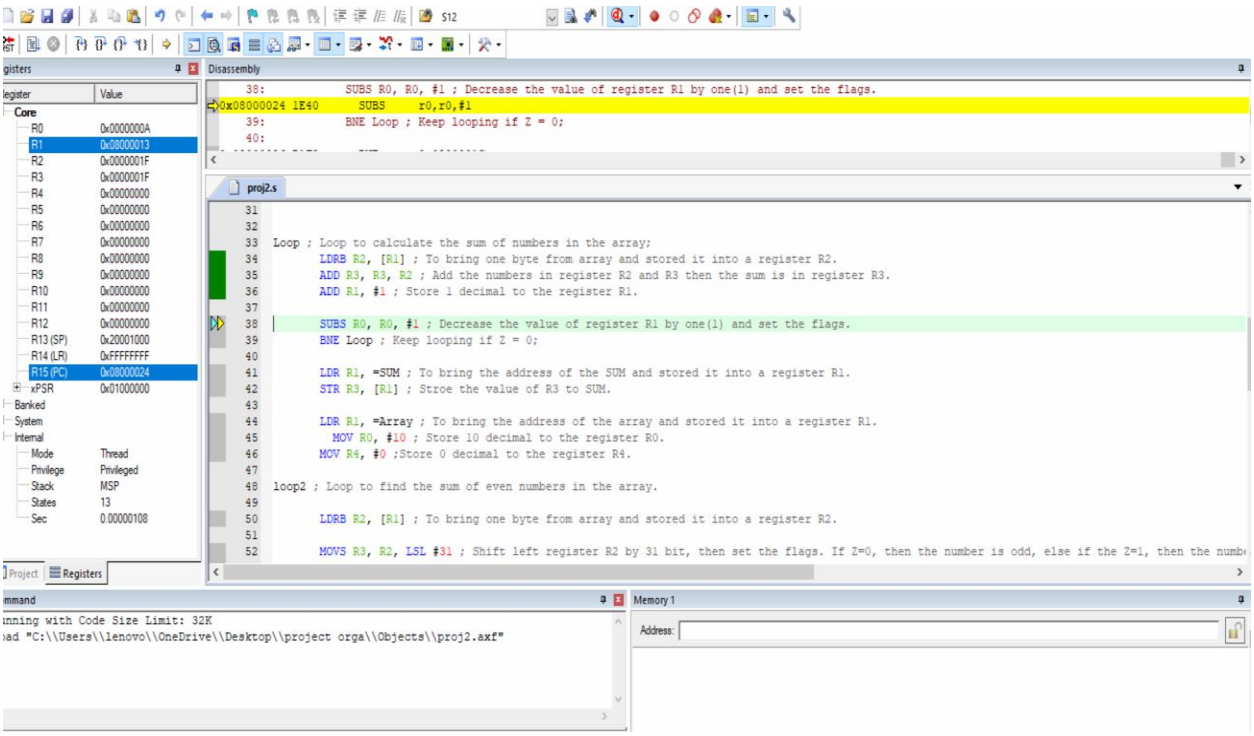
b) Find the sum of all elements of the array and store it in the memory, e.g. variable SUM.



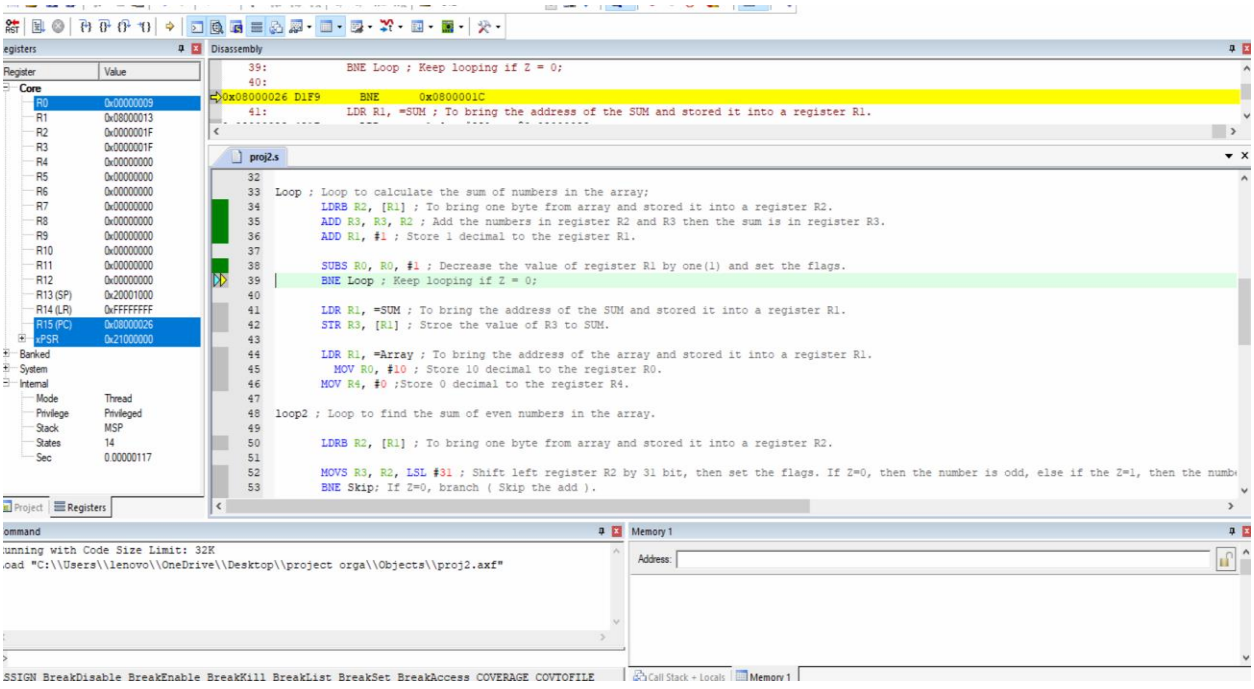
To bring one byte from array and stored it into a register R2.



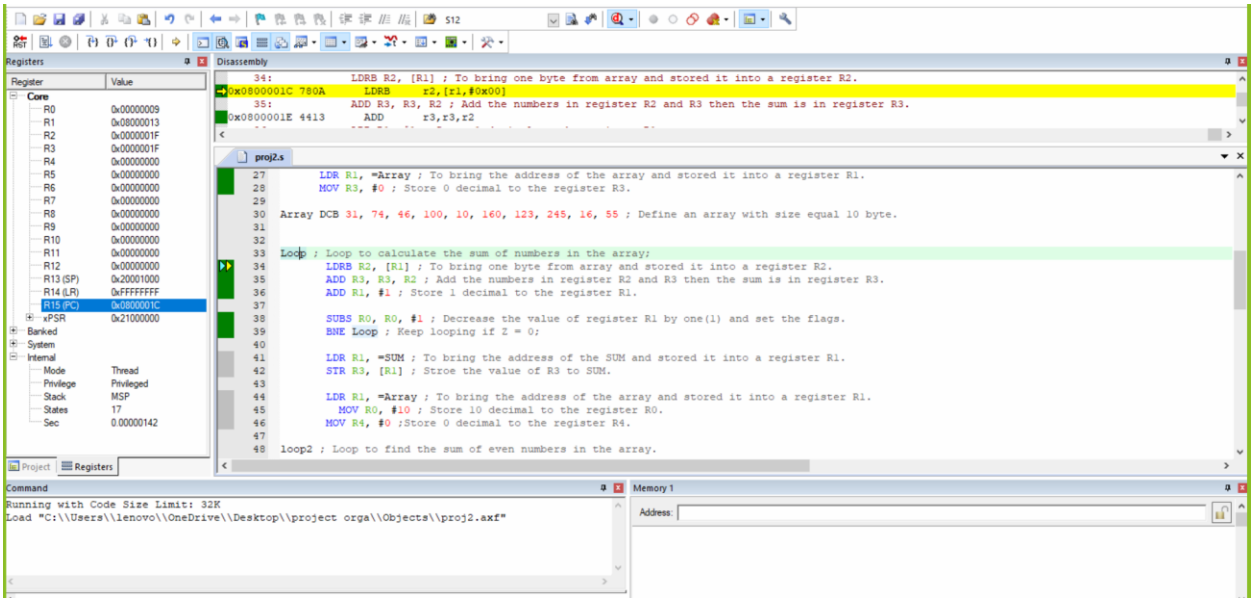
Add the numbers value in the register R2 with register R3.



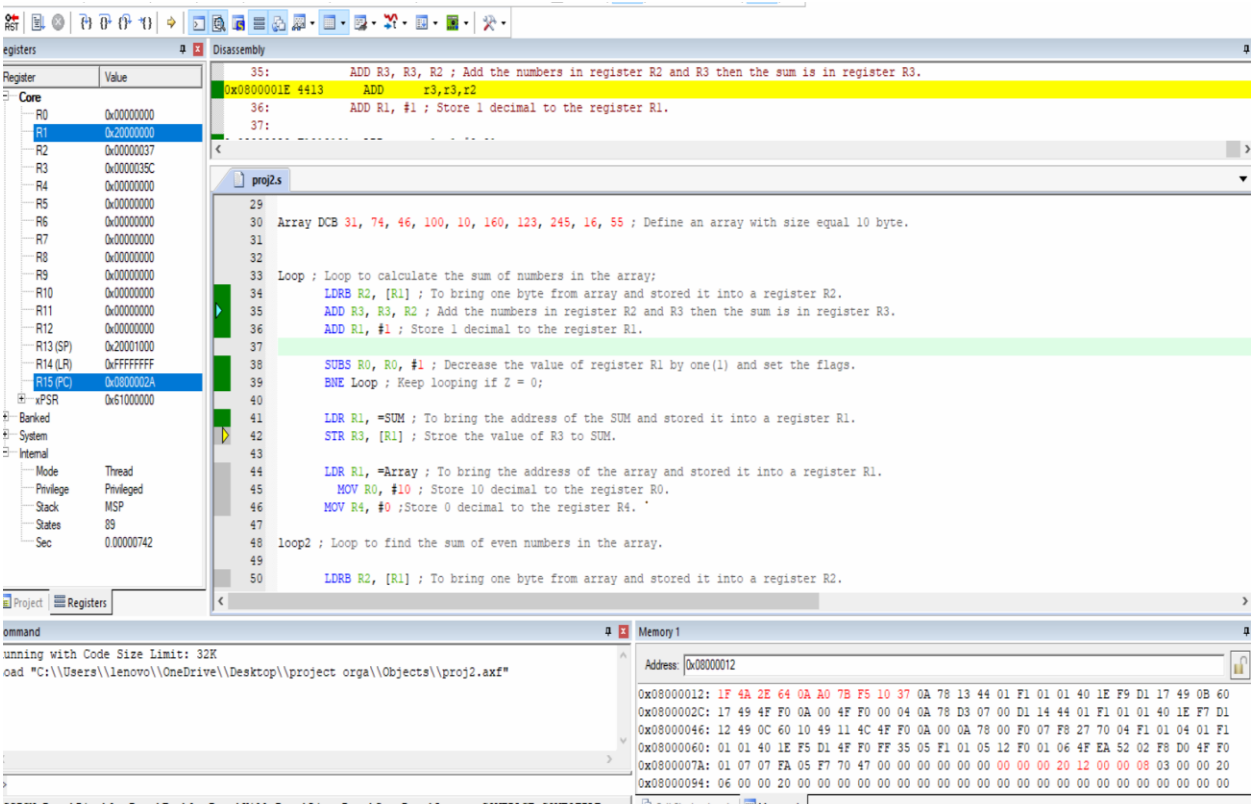
The register R1 indicate to the next index.



Decrease the value of register R1 by one.

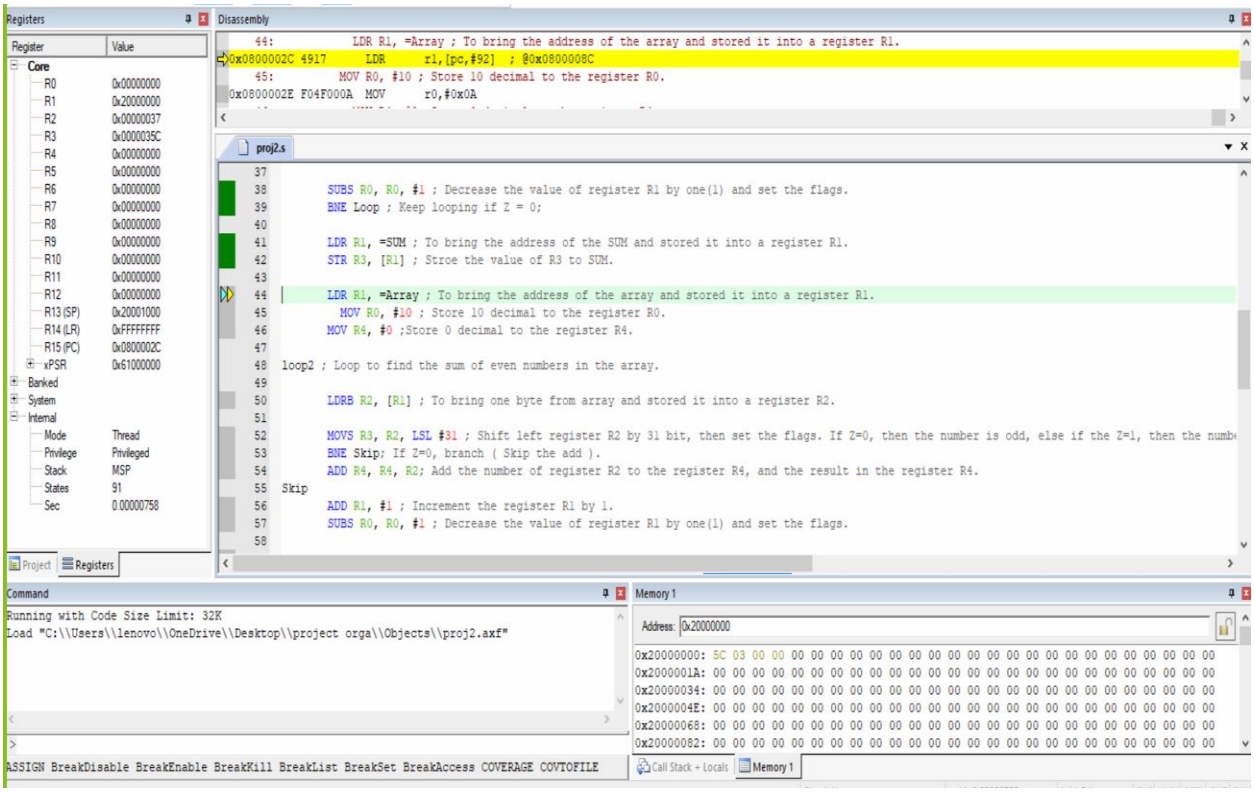


The loop repeated 10 times, and we adds the 10 numbers of the array.



The sum of array's values which it's stored to register R3 is **35C**

Stored the address of the SUM into a register R1.



We stored the value of register R3 to SUM.

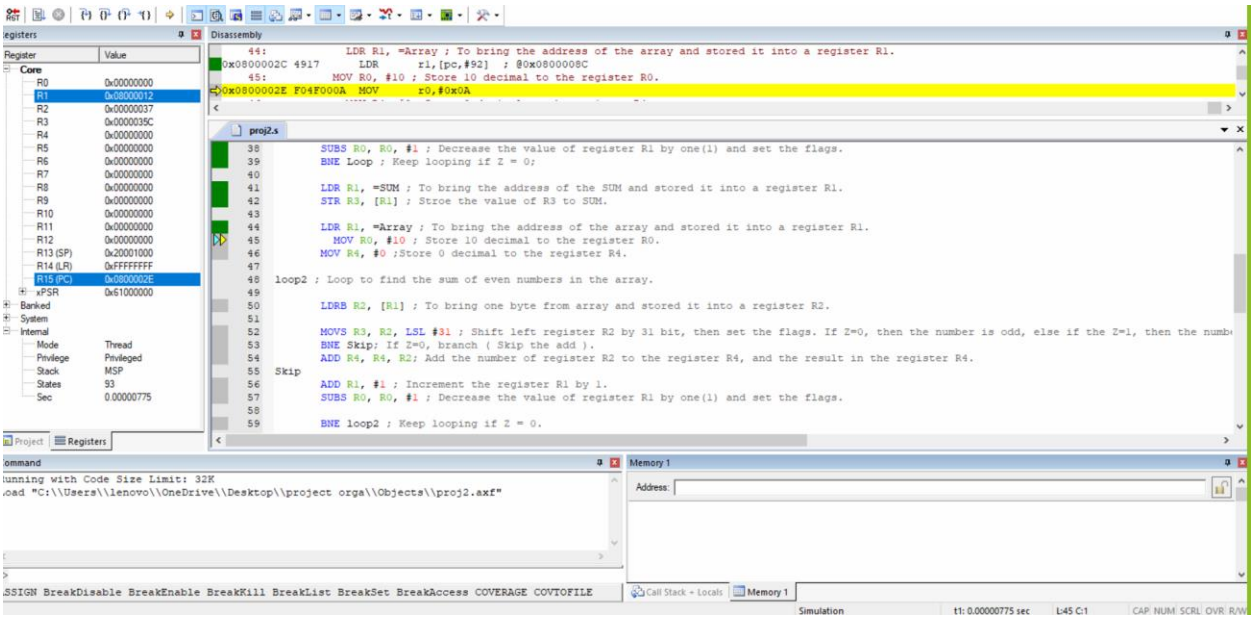
34+74+46+100+10+160+123+245+16+55=860

860 in decimal = 35C in hexadecimal which appear above.

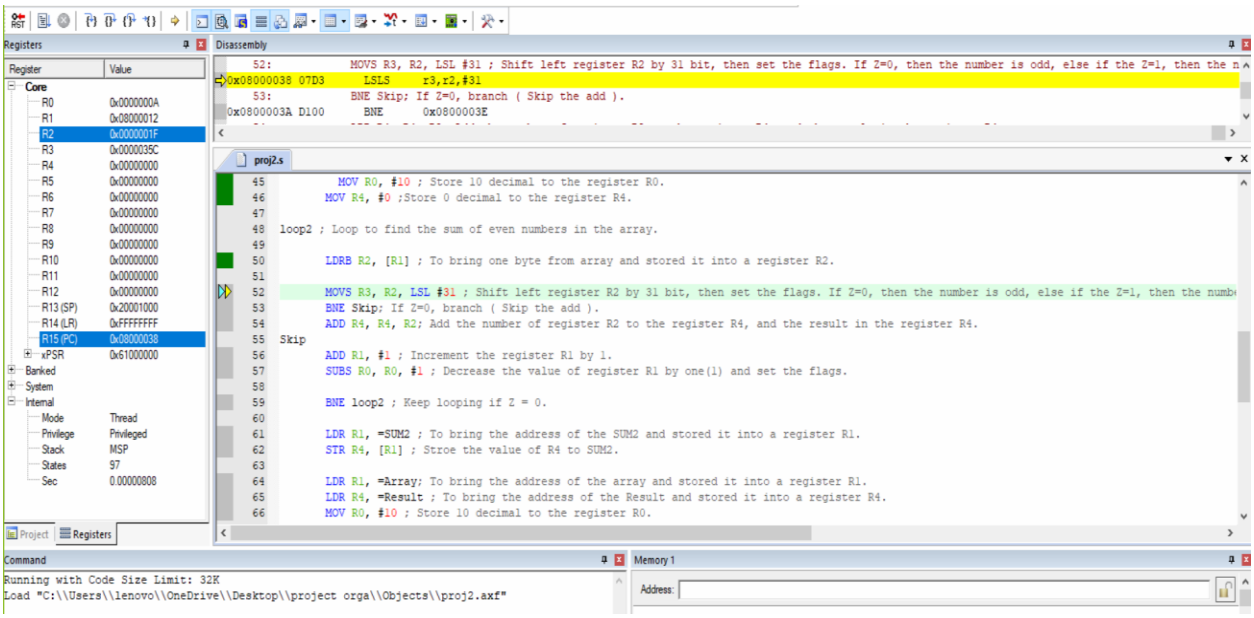
Underlined with blue color.

The second part is done

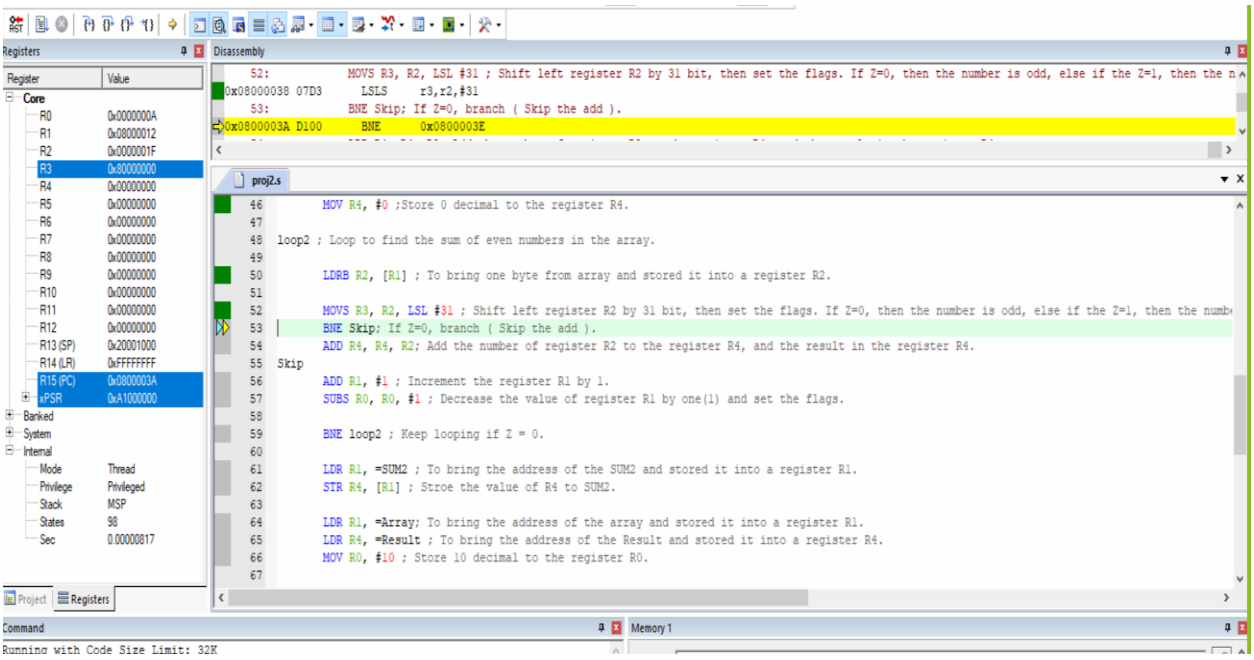
c) find the sum of the even numbers in this array and store it in the memory, e.g. variable EVEN



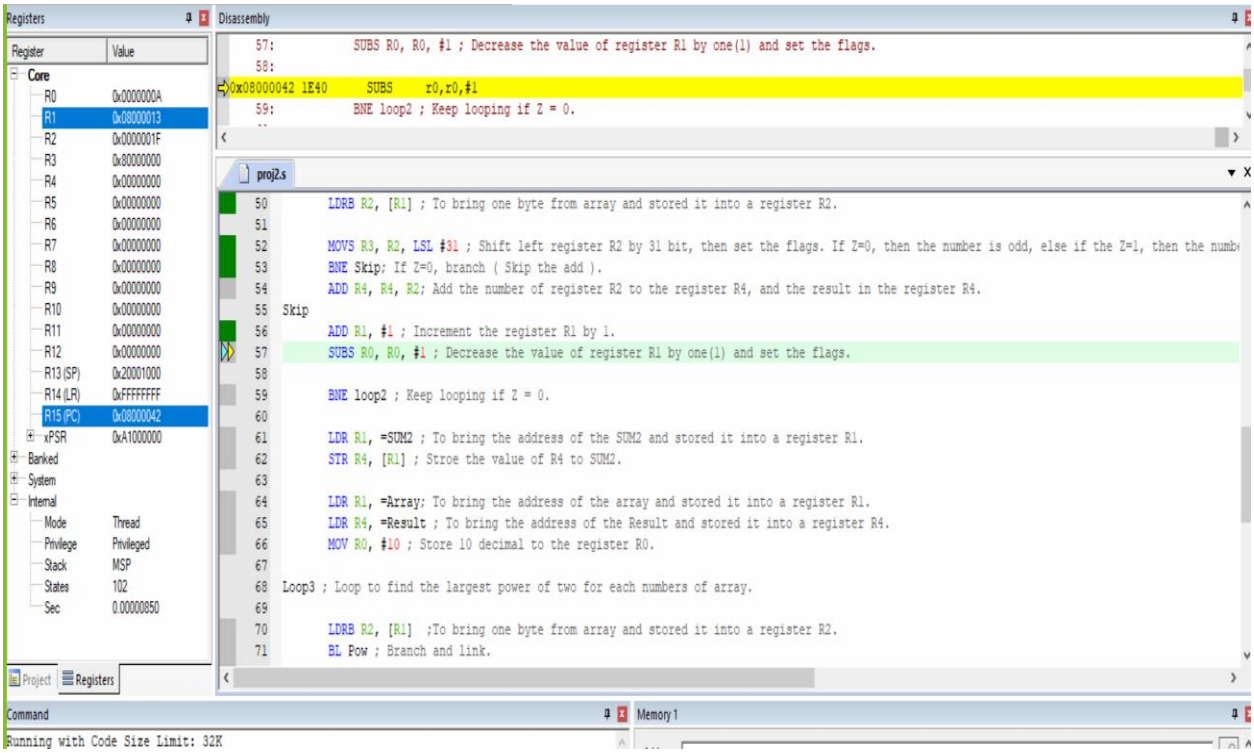
Store the address of the array in a register R1.



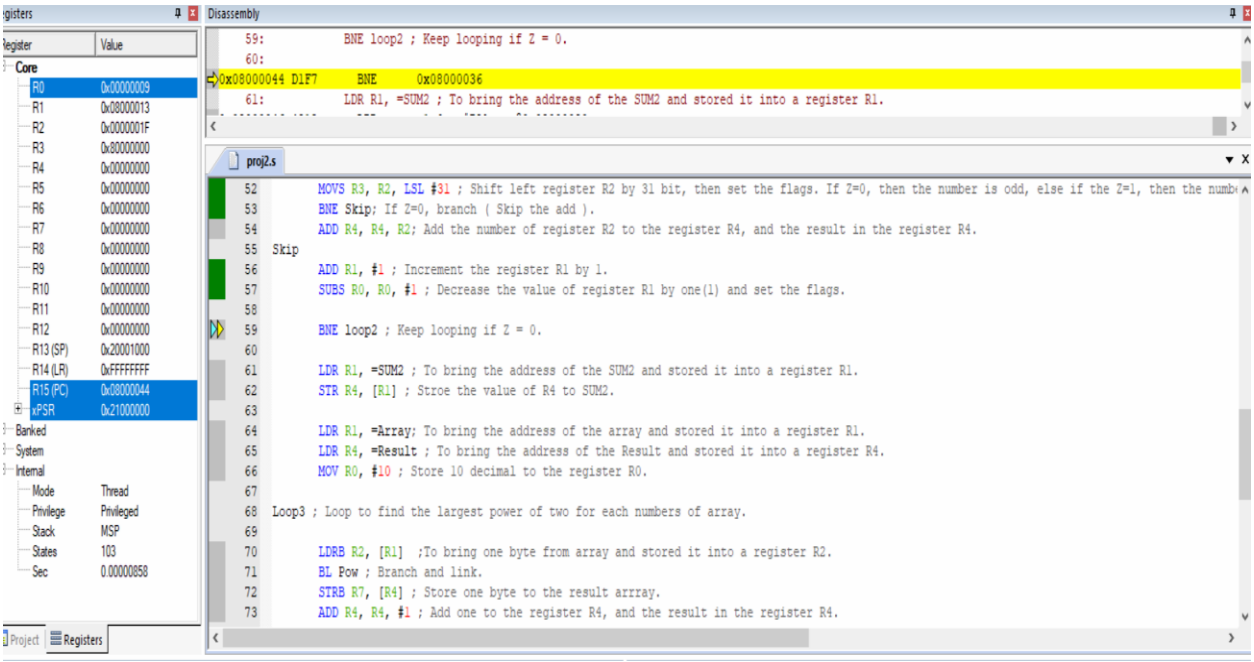
To bring one byte from array and stored it into a register R2.



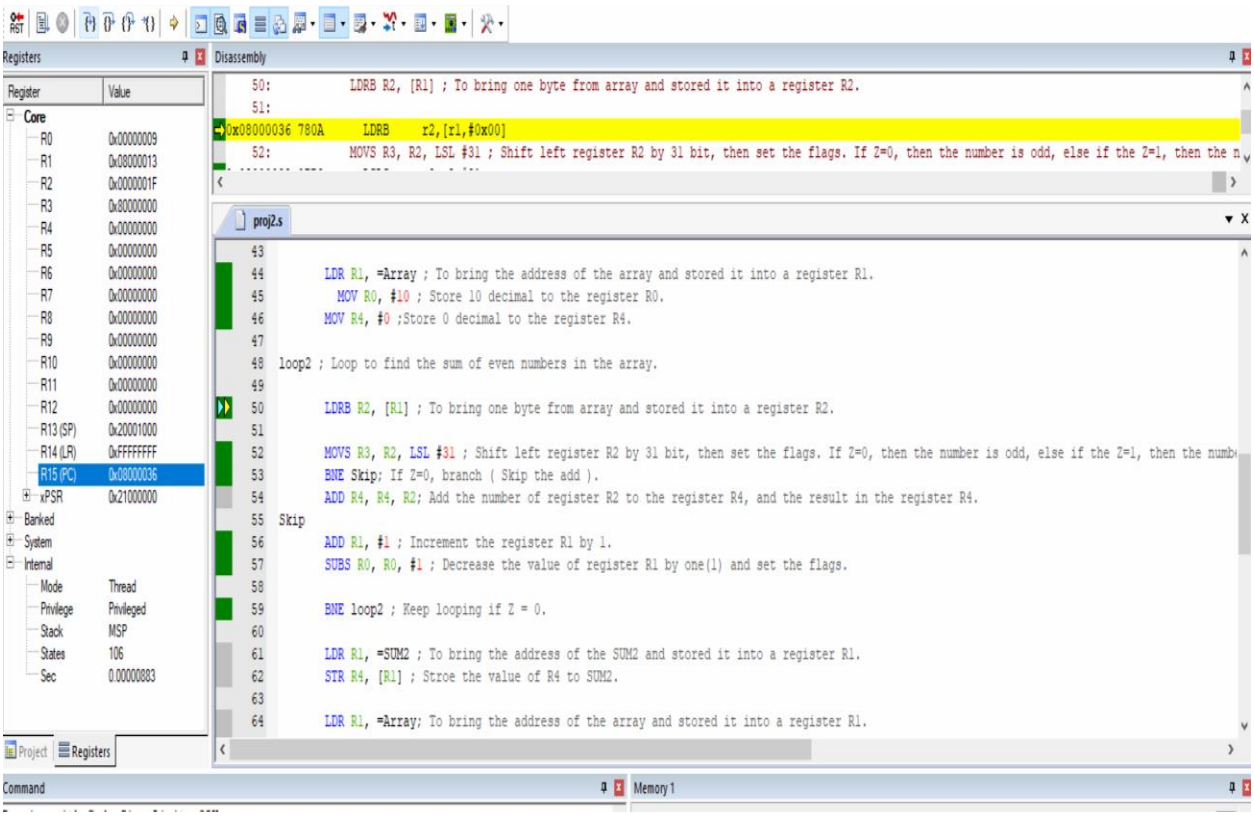
Shift left register R2 by 31 bit, then stored it into a register R3, then check if the number don't equal zero, and skip if it doesn't equal zero.



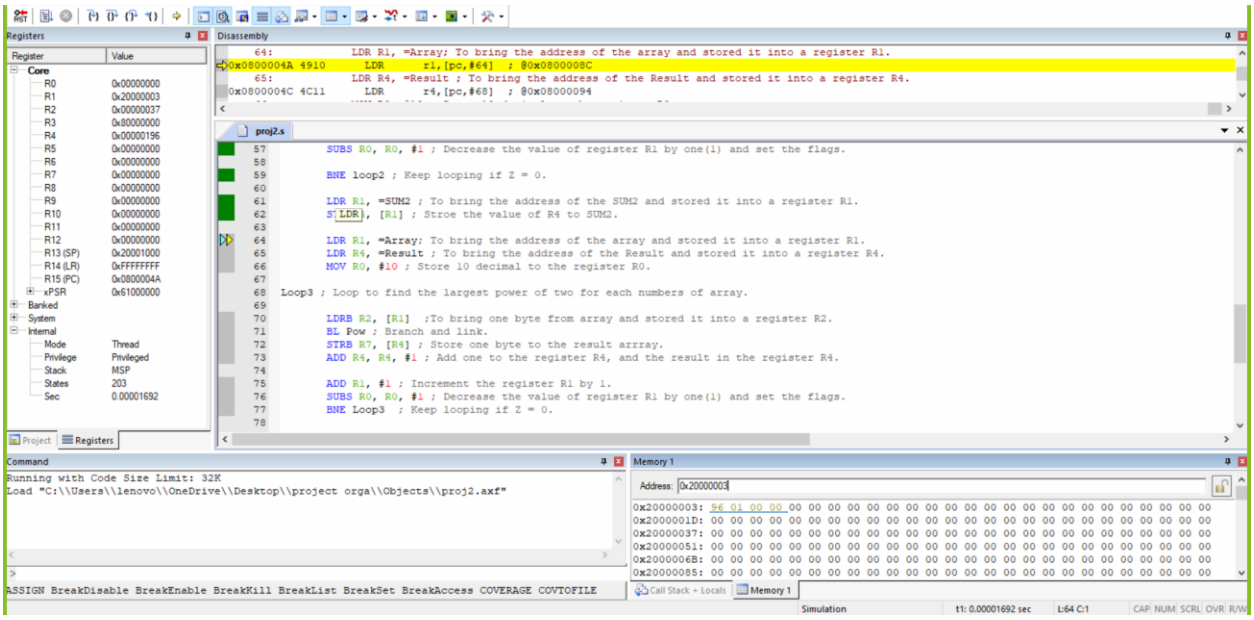
The register R1 indicate to the next index.



Decrease the value of register R1 by one.

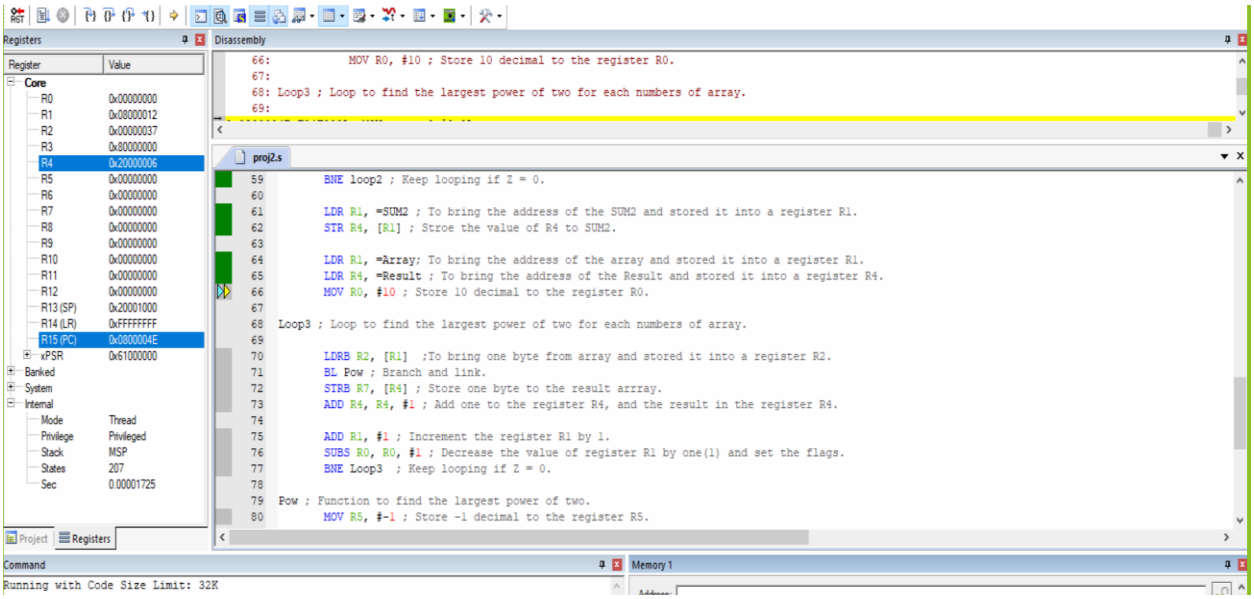


Repeated the loop 10 times.



Stored the sum of even numbers in a SUM2

Even numbers: 74+46+100+10+160+16= 406 in decimal which equal 196 in hexadecimal and this appeared above, underlined with blue color.

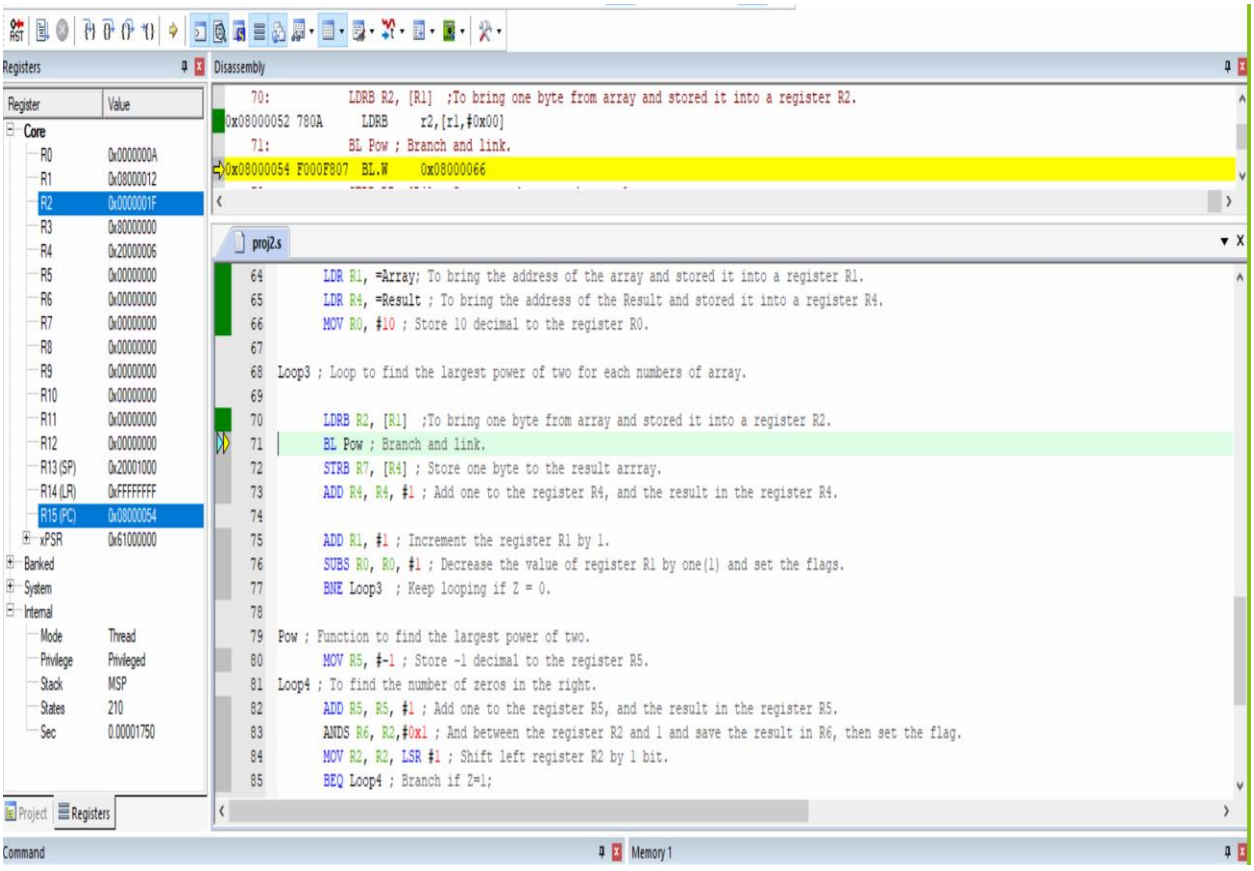


Stored the address of the array into a register R1, and stored the address of the result into a register R4.

The third Part is done

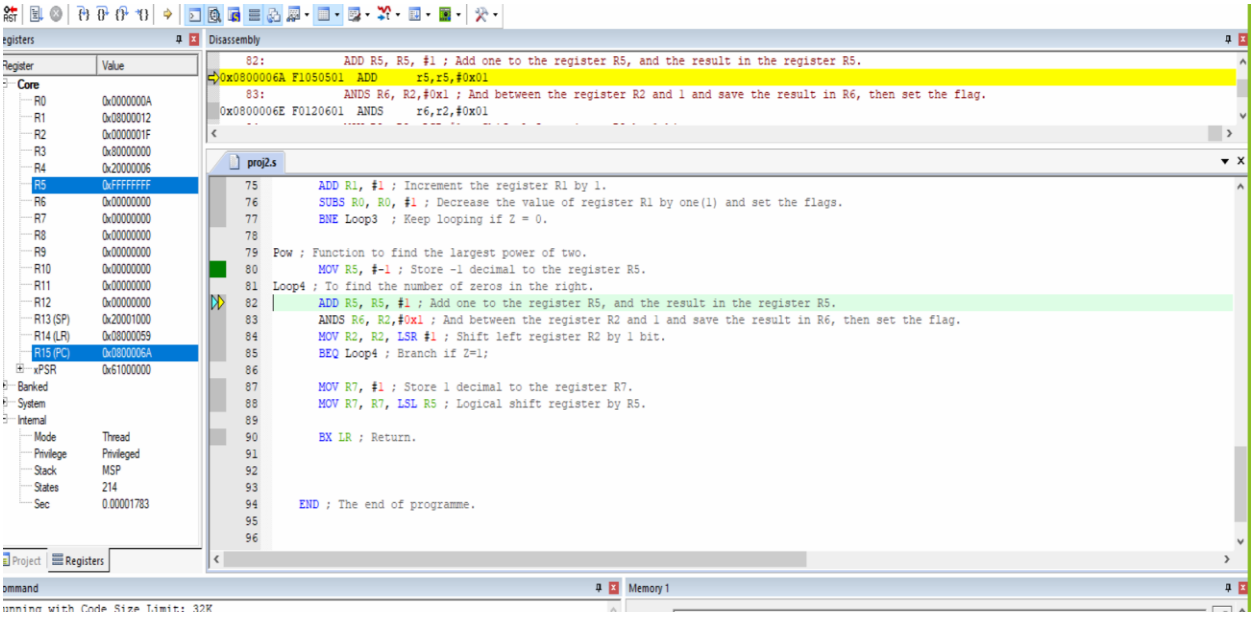
d) Find the largest power of 2 divisor that divides into a number exactly for each element in the array and store it in another array in the memory. You have to use a procedure (function), POW2, which takes an integer as an input parameter and return its largest power of 2. For example, POW(52) would return 4, where POW(56) would return 8, and so on.

Hint: You can find the largest power of 2 dividing into a number exactly by finding the rightmost bit of the number. For example, (52)10 = (110100)2 __ has its rightmost bit in the 4's place, so the largest power of 2 divisor is 4; (56)10 = (111000)2 has the rightmost bit in the 8's place, so its largest power of 2 divisor is 8

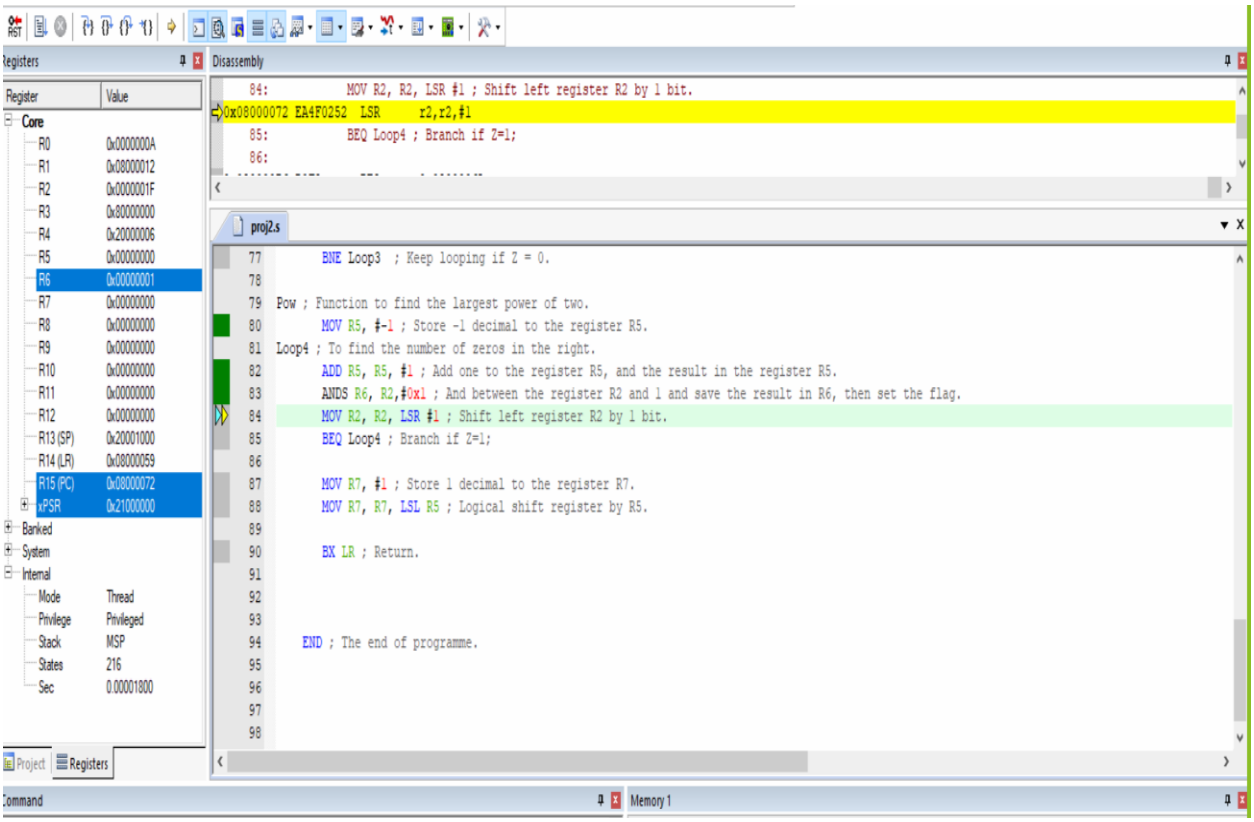


To bring one byte from array and stored it into a register R2.

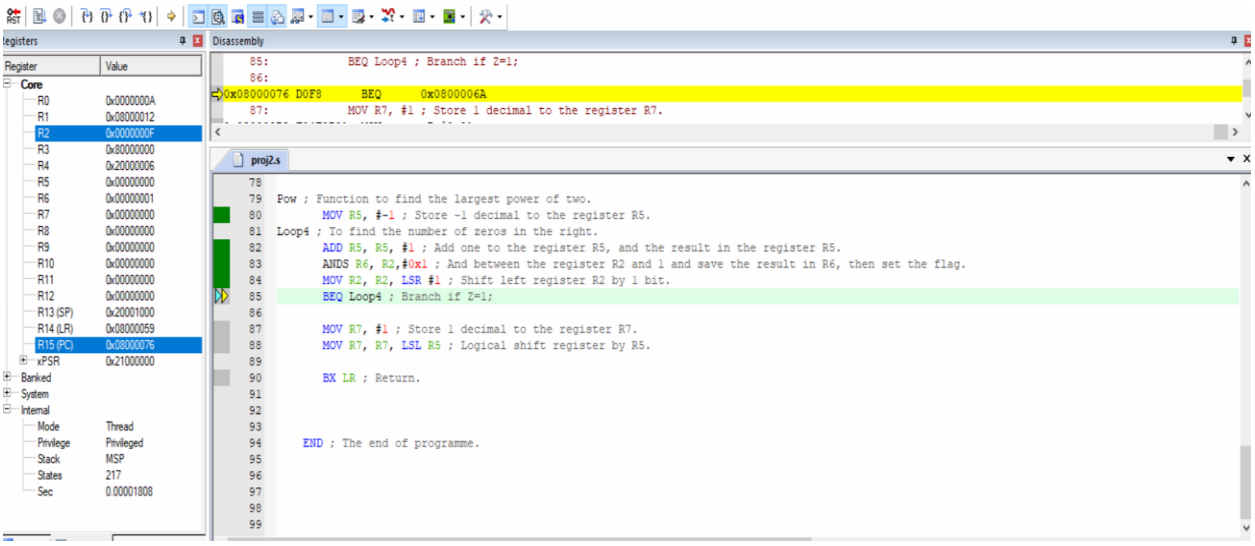
Branch and link.



Stored -1 decimal to the register R5.

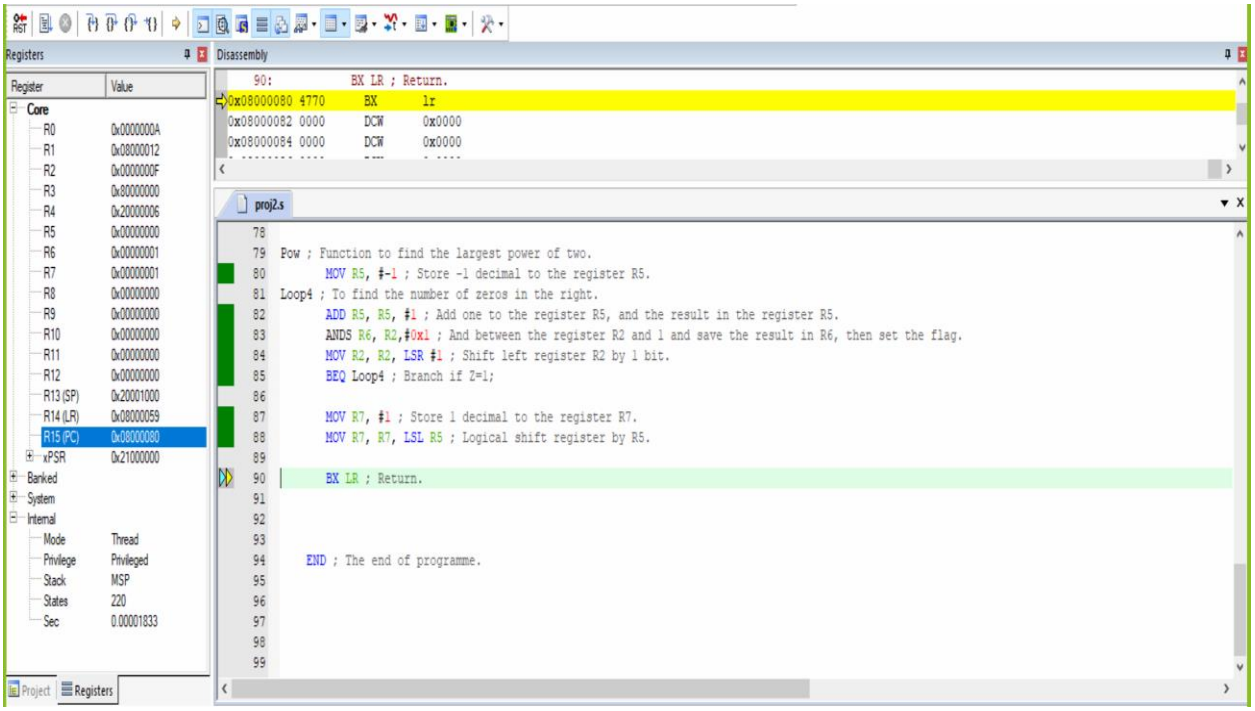


And between the register R2 and 1.

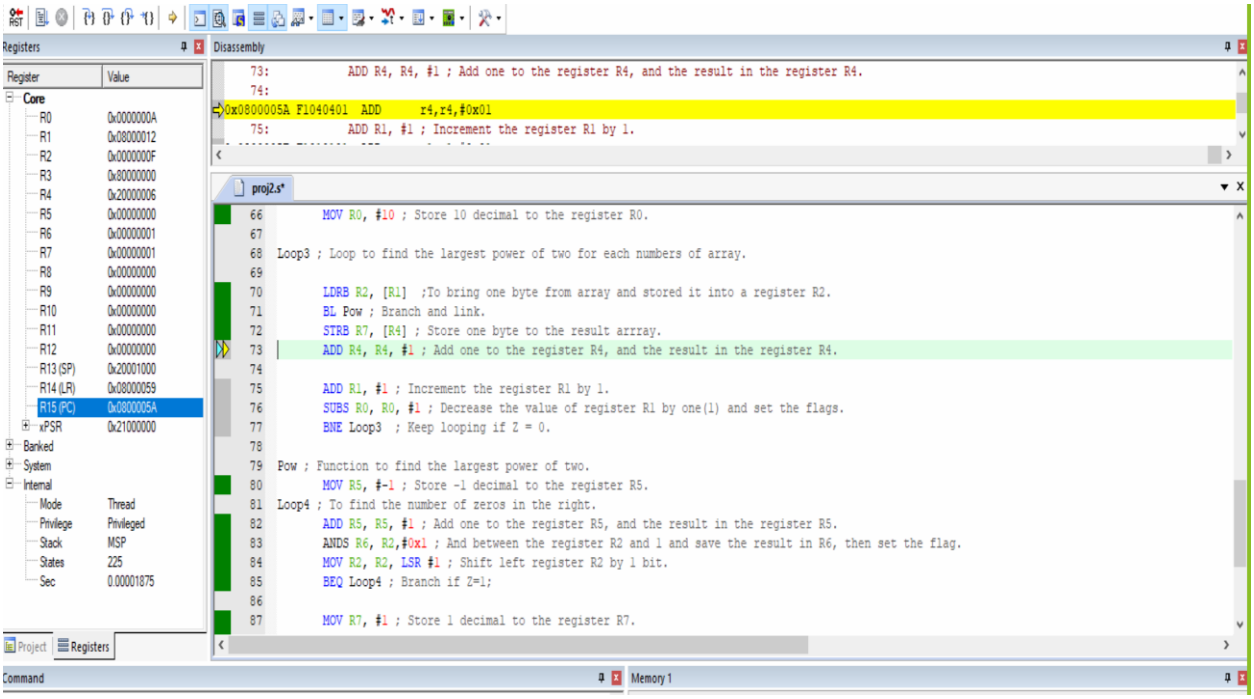


Shirt right register R2 by 1 bit. (left in the code is a mistake in writing)

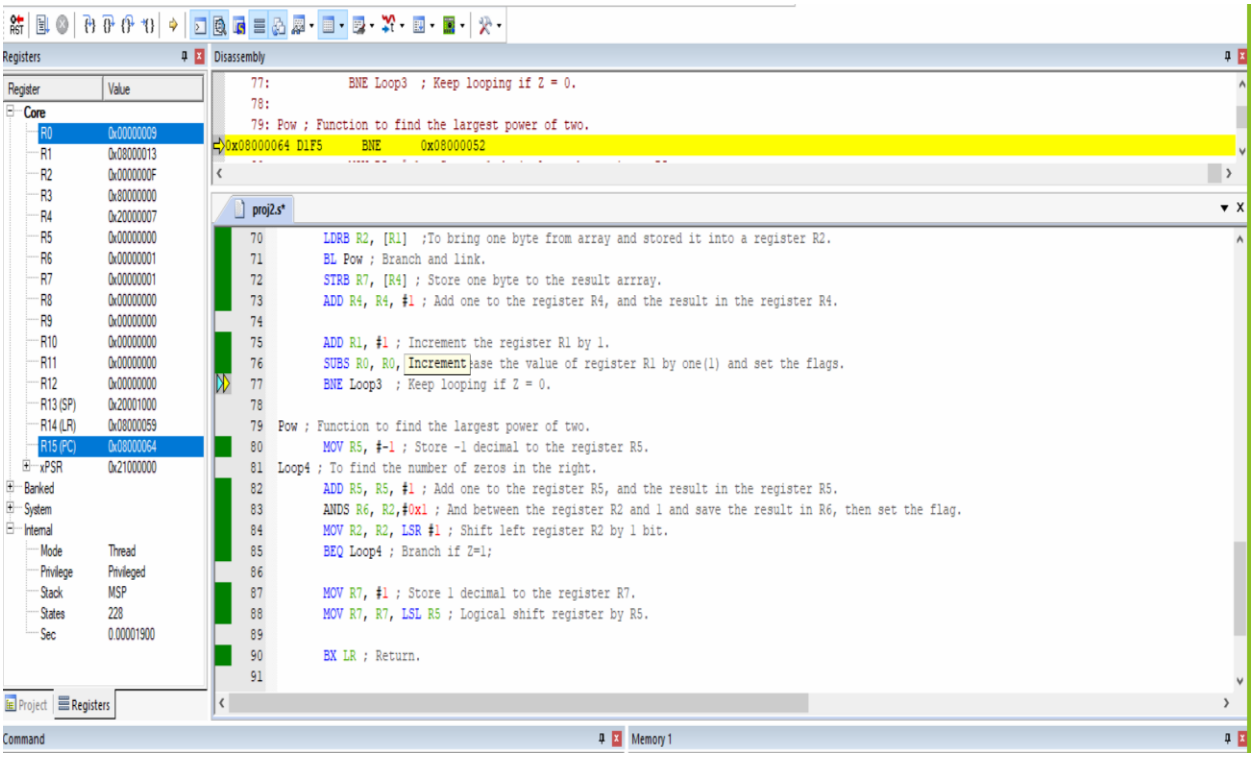
And repeated the loop to become the first index equal 1.



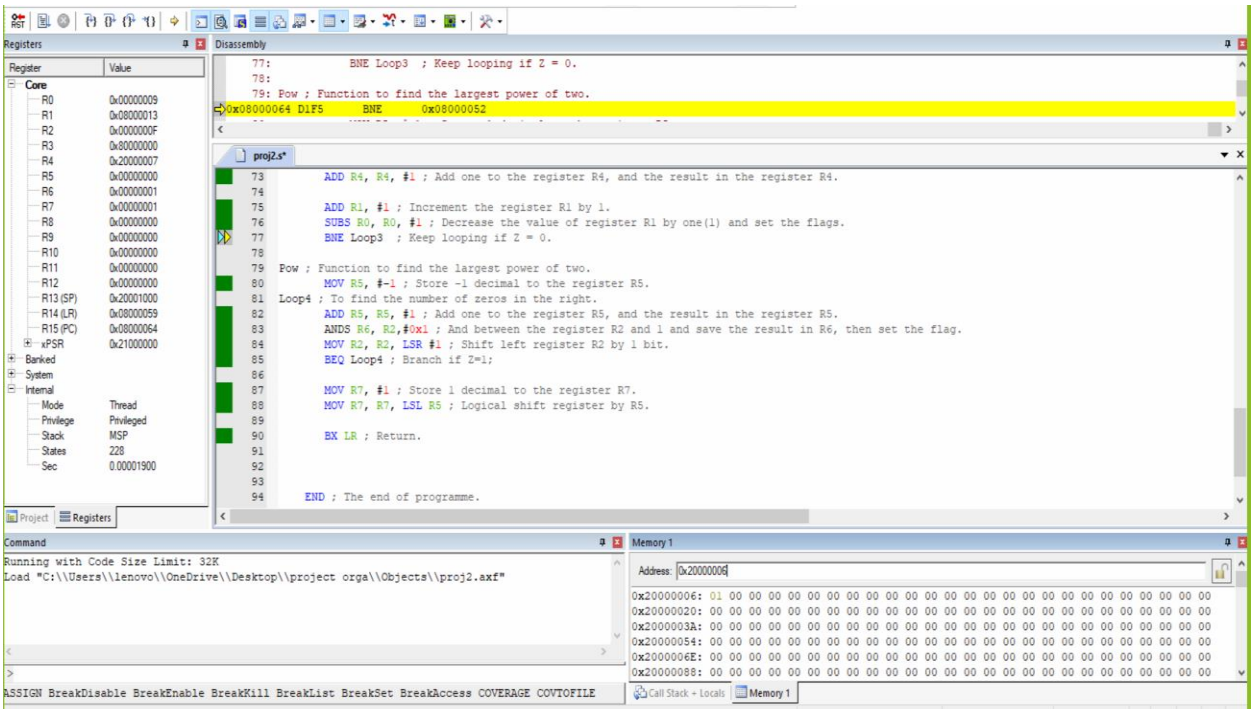
Shift left register R7 by the register R5 value’s, then return.



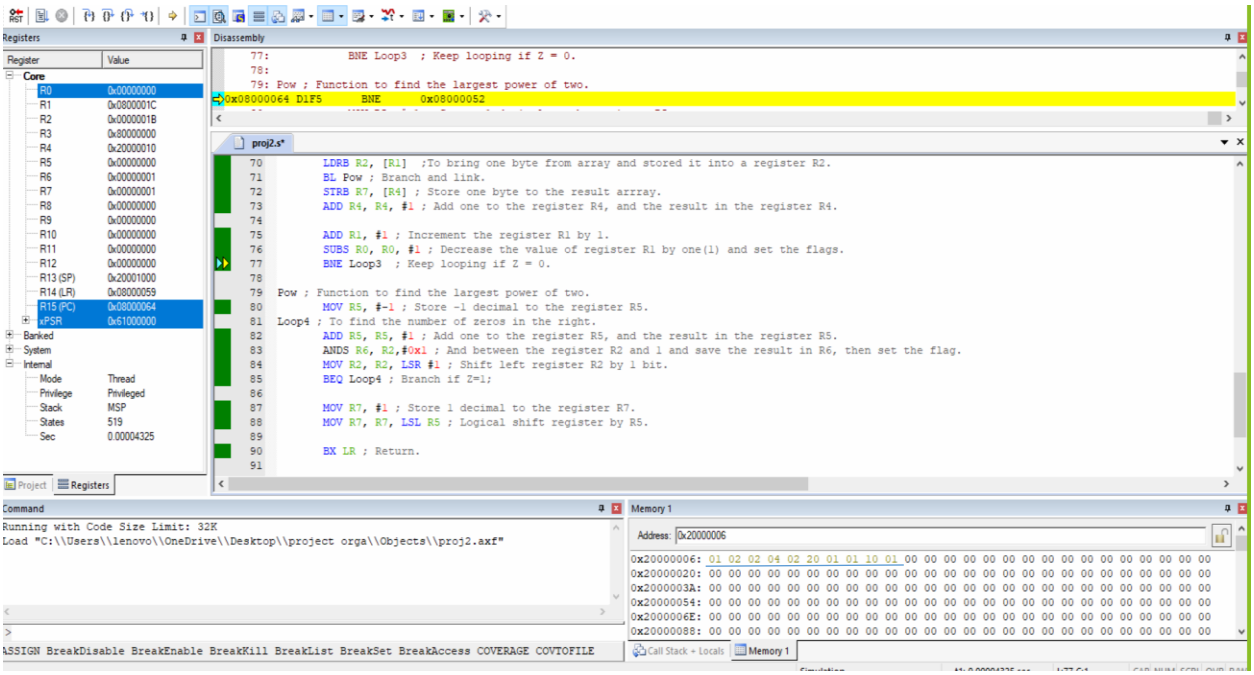
Stored the value of register R7 into a Result.



The register R1 and R4 indicates to the next index.



Stored the Result to the new array, and repeated the loop 10 times to include all numbers.



The outputs stored in the variable Result, underlined with blue color.

1- POW(31)(11111) = 2^0= 01

2- POW(74)(1001010) = 2^1=02

3- POW(46)(101110) =2^1=02

4- POW(100)(1100100)=2^2=04

5- POW(10)(1010)=2^1=02

6- POW(160)(10100000)=(32 in decimal) = 20 in hexadecimal.

7- POW(123)(1111011)=2^0=01

8- POW(245)(11110101)=2^0=01

9- POW(16)(10000)=(16 in decimal)=10 in hexadecimal.

10- POW(55)(110111)=2^0=01

The fourth part is done 

The assignment is done

Thank you Dr. Ahmad Afaneh