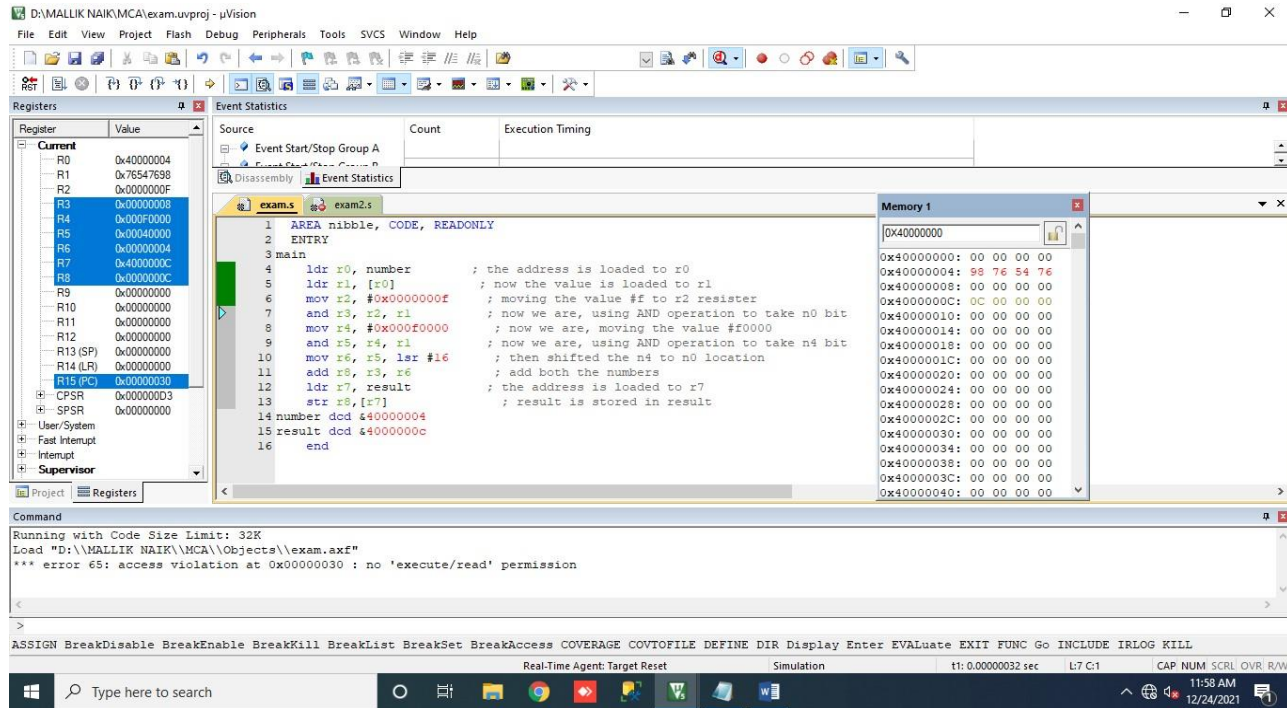
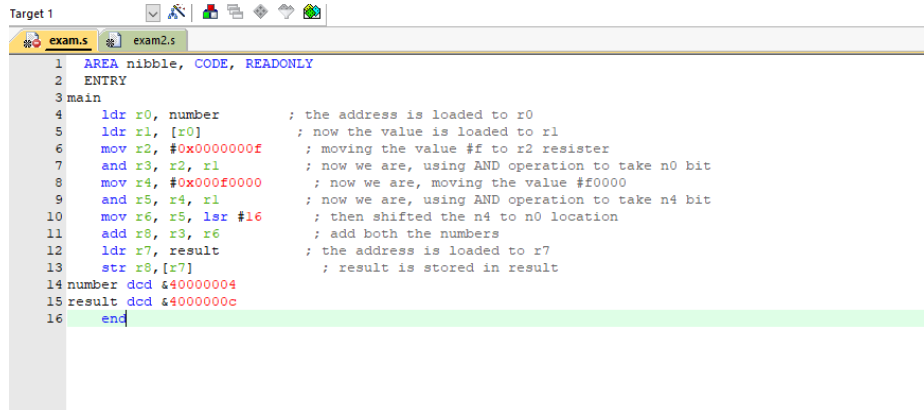


NAME:- MALLIK THANU NAIK  
ROLL NUMBER :- 211039038  
1<sup>ST</sup> QUESTION adding Nibble



## PROGRAM FOR THE FIRST ONE



## 2<sup>ND</sup> QUESTION positive number addition

The screenshot shows the uVision IDE interface. The main window displays the assembly code for a program named 'exam2.s'. The code is as follows:

```
1 AREA arrt, CODE, READONLY
2 ENTRY
3 main
4 mov r0, #00000004 ; move the number of element in the array
5 ldr r1, array ;loading the array address to the r1
6 loop cmp r0, #0 ; compare it with r0 i.e count value
7 beq stop ; this will end the loop
8 sub r0, r0, #1 ; the count is keep on decrementing
9 ldr r3, [r1, #4]! ;taking the value from the address
10 cmp r3, #0 ; comparing the value with zero
11 bpl counts ; if it is positive then it will add
12 bmi result ; if it is negative then it will not add
13 counts add r2, r2, r3 ; the r2 value and r3 value is added
14 result ldr r5, count ; the address is stored in r5
15 STR r2, [r5] ; the sum will be repetedly stored in the r2
16 b loop
17
18 stop b stop ; this is if the lopp will end
```

The registers window on the left shows the current state of the processor registers. The command window at the bottom shows the program running with a code size limit of 32K. The status bar at the bottom indicates the program is running in simulation mode.

## PROGRAM FOR THE SECOND ONE

```
1 AREA arrt, CODE, READONLY
2 ENTRY
3 main
4 mov r0, #00000004 ; move the number of element in the array
5 ldr r1, array ;loading the array address to the r1
6 loop cmp r0, #0 ; compare it with r0 i.e count value
7 beq stop ; this will end the loop
8 sub r0, r0, #1 ; the count is keep on decrementing
9 ldr r3, [r1, #4]! ;taking the value from the address
10 cmp r3, #0 ; comparing the value with zero
11 bpl counts ; if it is positive then it will add
12 bmi result ; if it is negative then it will not add
13 counts add r2, r2, r3 ; the r2 value and r3 value is added
14 result ldr r5, count ; the address is stored in r5
15 STR r2, [r5] ; the sum will be repetedly stored in the r2
16 b loop
17
18 stop b stop ; this is if the lopp will end
19 array dcd 40000000
20 count dcd 40000000
21 end
```