Summary

-For Q1, to tackle the problem the program counts the times of carries for a number.
(i) 0x12345678 *returns 1 (odd)
(ii) 0xF0F0F0F0 *returns 0 (even)
(iii) 42 *returns 1 (odd)
The returned values were that of expected. The program for Q1 works.

-For Q2, algorithm 1, I translated the pseudo code into assembly.

Question	Numerator	Divisor	R0 Quotient	R1 Remainder
i	27	7	0x3	0x6
ii	444444	23	0x4B7B	0xF
iii	33554432	506	0x10309	0x36

The returned values were that of expected. The program for Q2 algorithm 1 works.

-For Q2, algorithm 2, I translated the pseudo code into assembly.

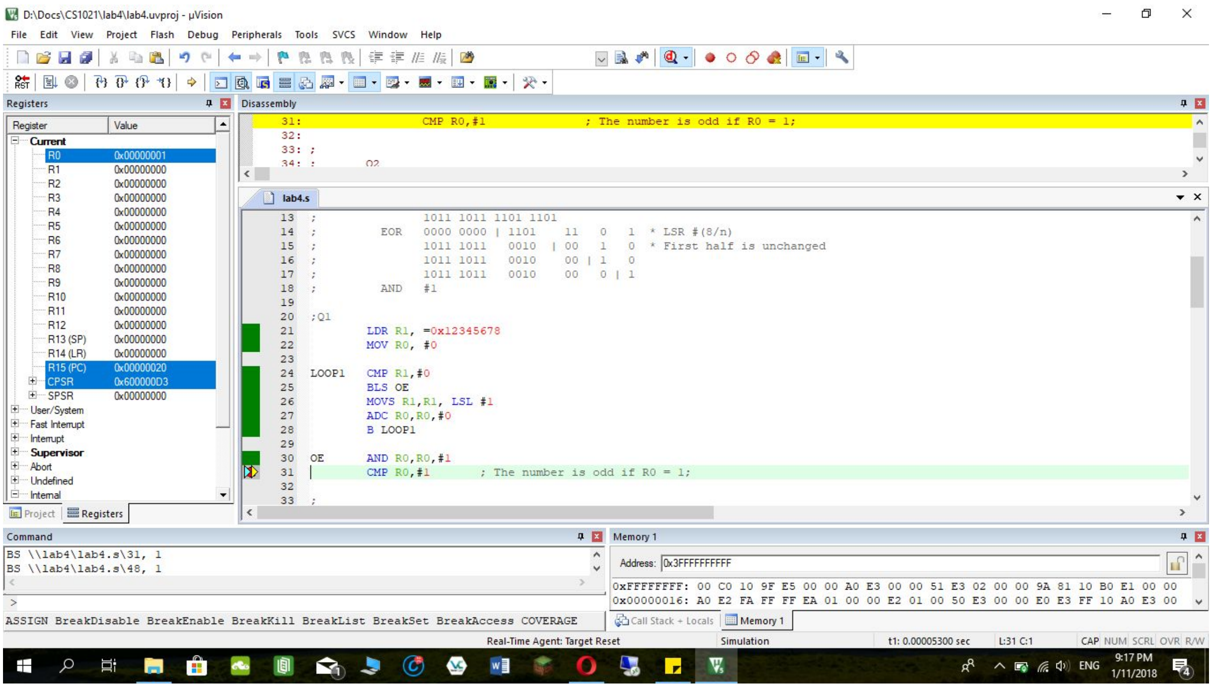
Question	Numerator	Divisor	R0 Quotient	R1 Remainder
i	27	7	0x3	0x6
ii	444444	23	0x4B7B	0xF
iii	33554432	506	0x10309	0x36

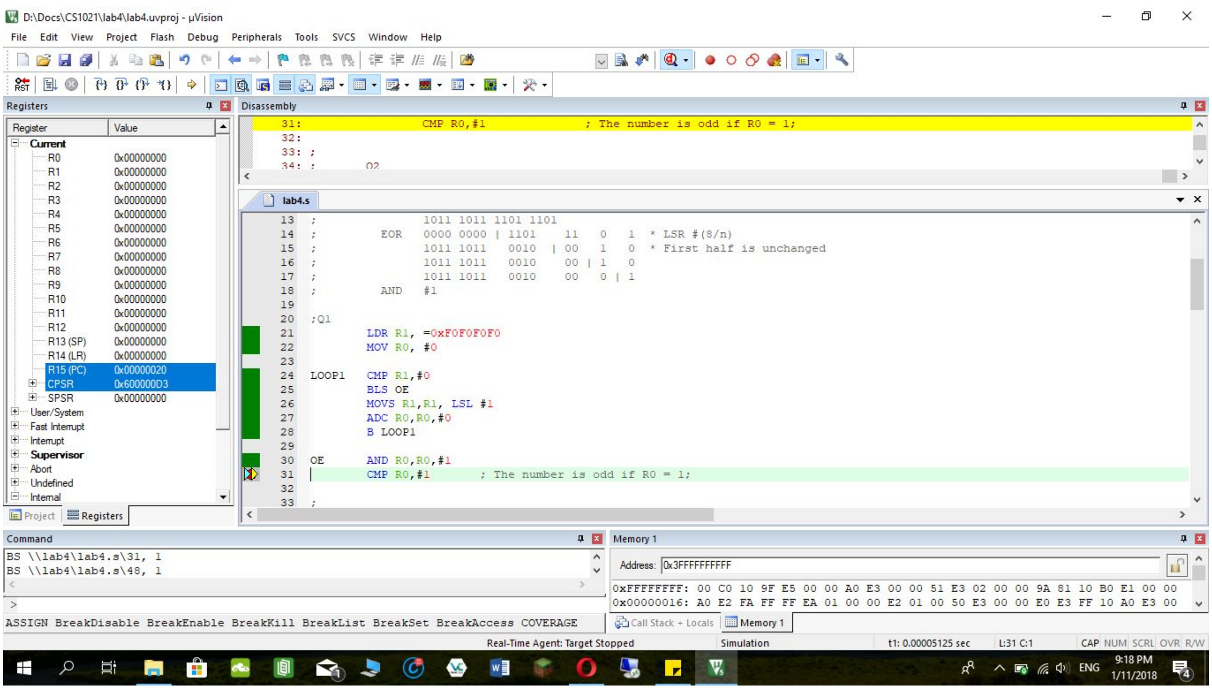
The returned values were that of expected. The program for Q2 algorithm 2 works.

Name: YI XIANG TAN
Student ID:18315028

Evidence

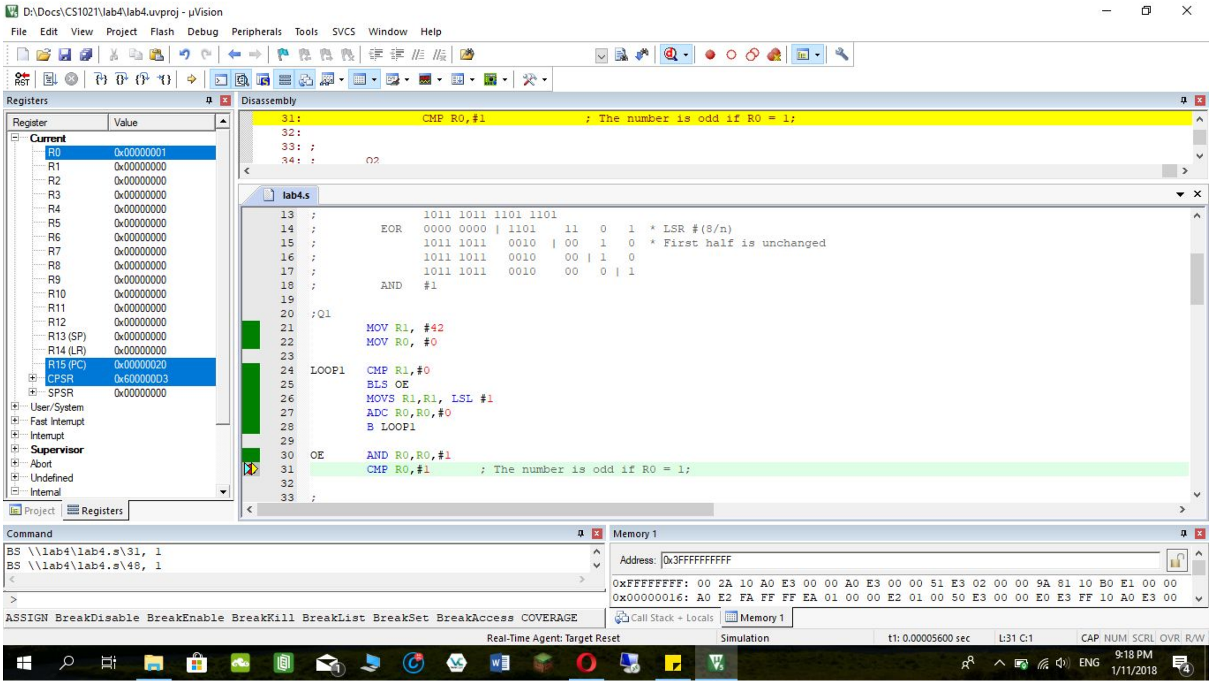
Each of the pictures is a screenshot of the registry for Q1(i),Q1(ii),Q1(iii),Q2A1(i),Q2A1(ii),Q2A1(iii),Q2A2(i),Q2A2(ii) and Q2A2(iii) respectively.

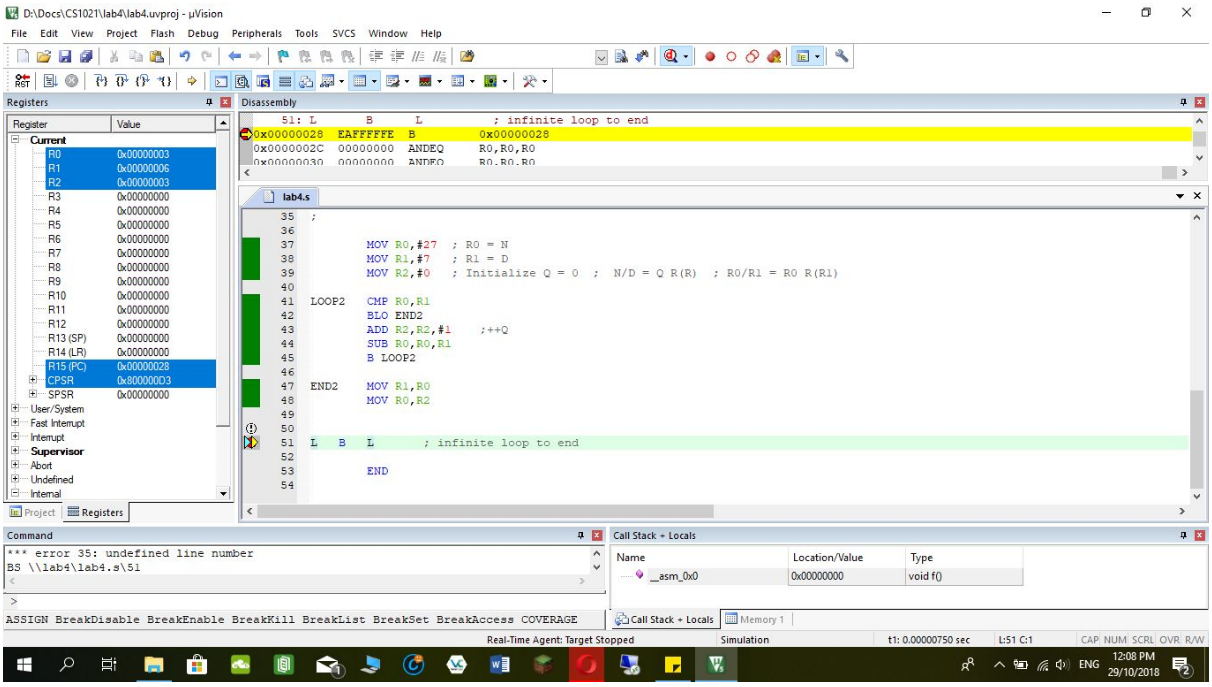




Name: YI XIANG TAN
Student ID:18315028

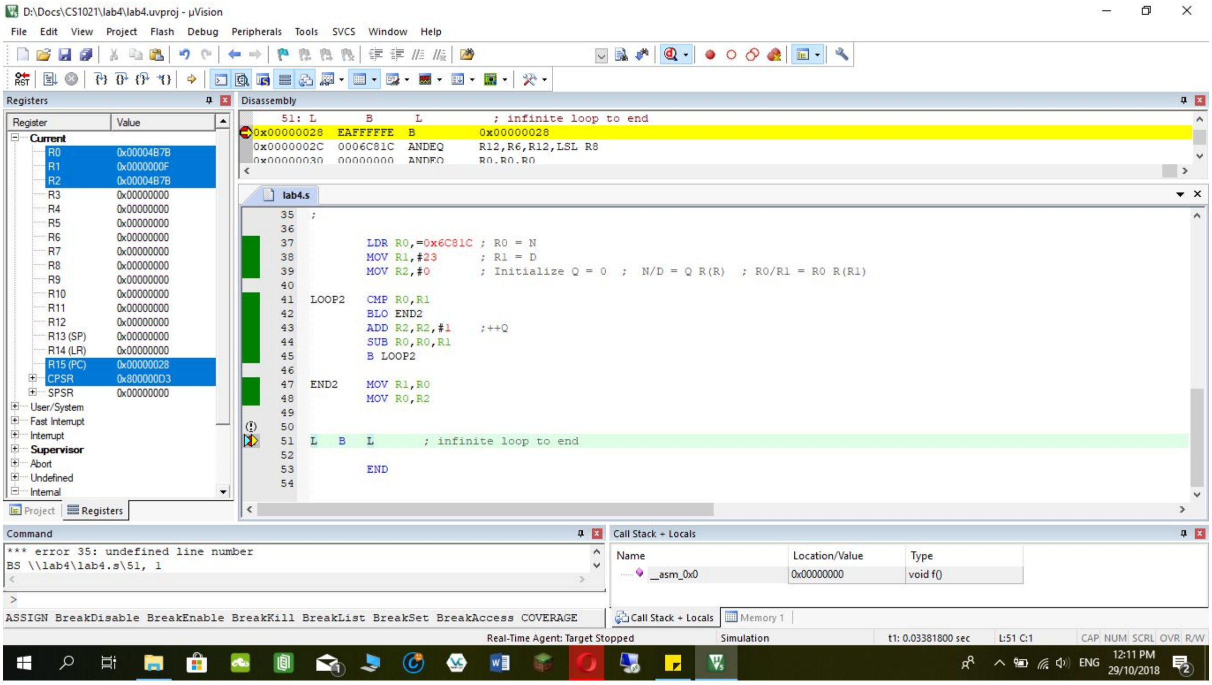
CS1021 Lab 4 Report

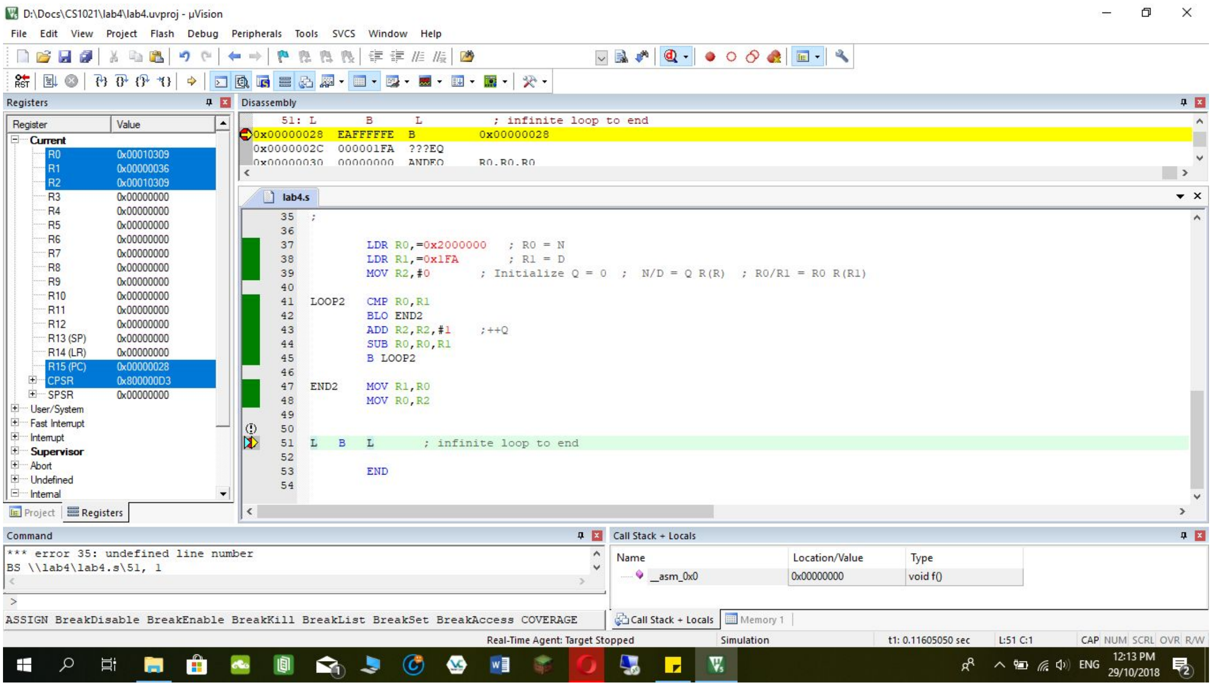




Name: YI XIANG TAN
Student ID:18315028

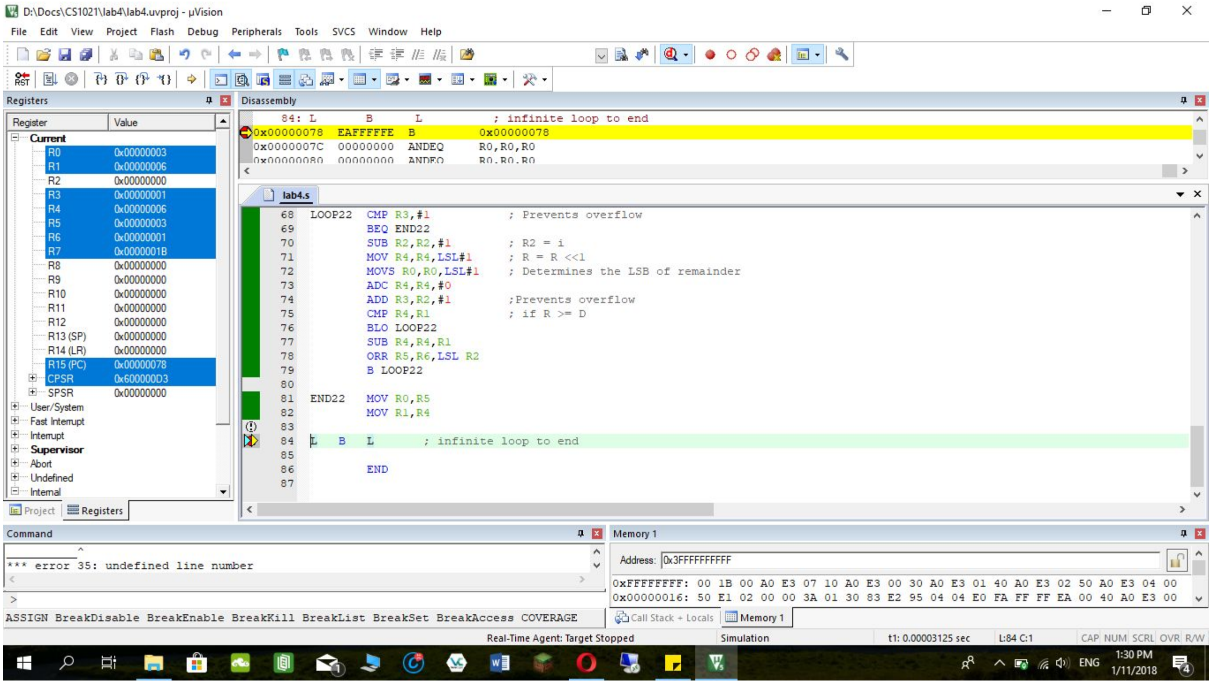
CS1021 Lab 4 Report



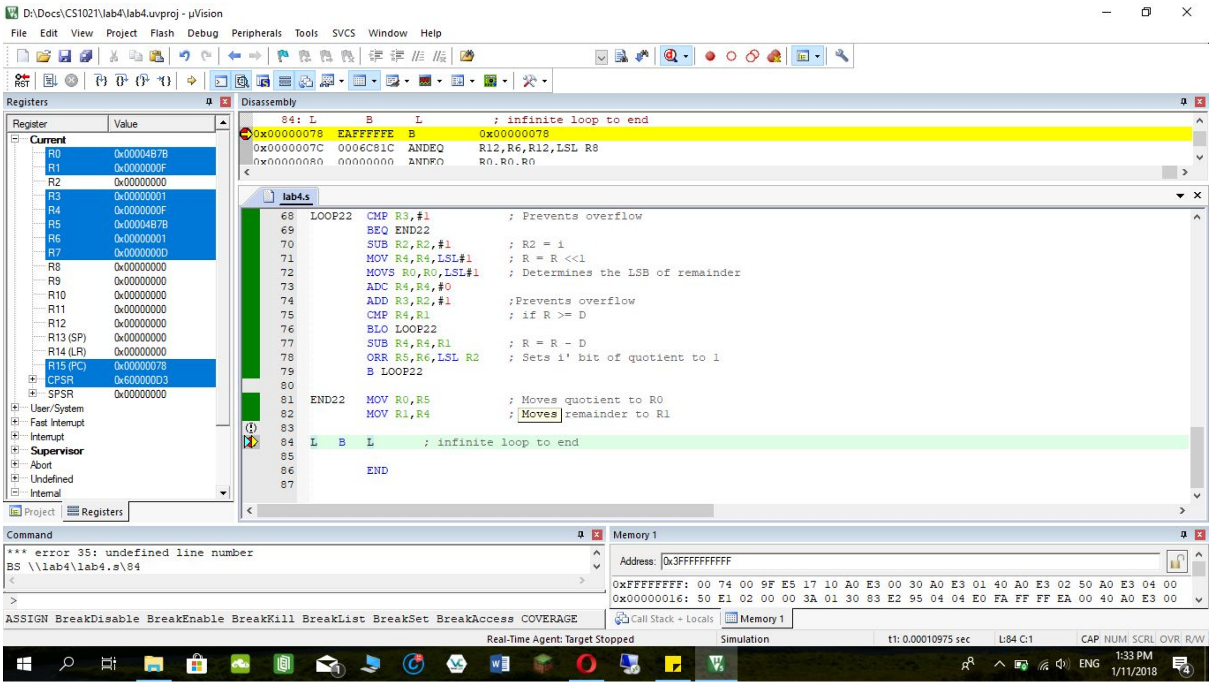


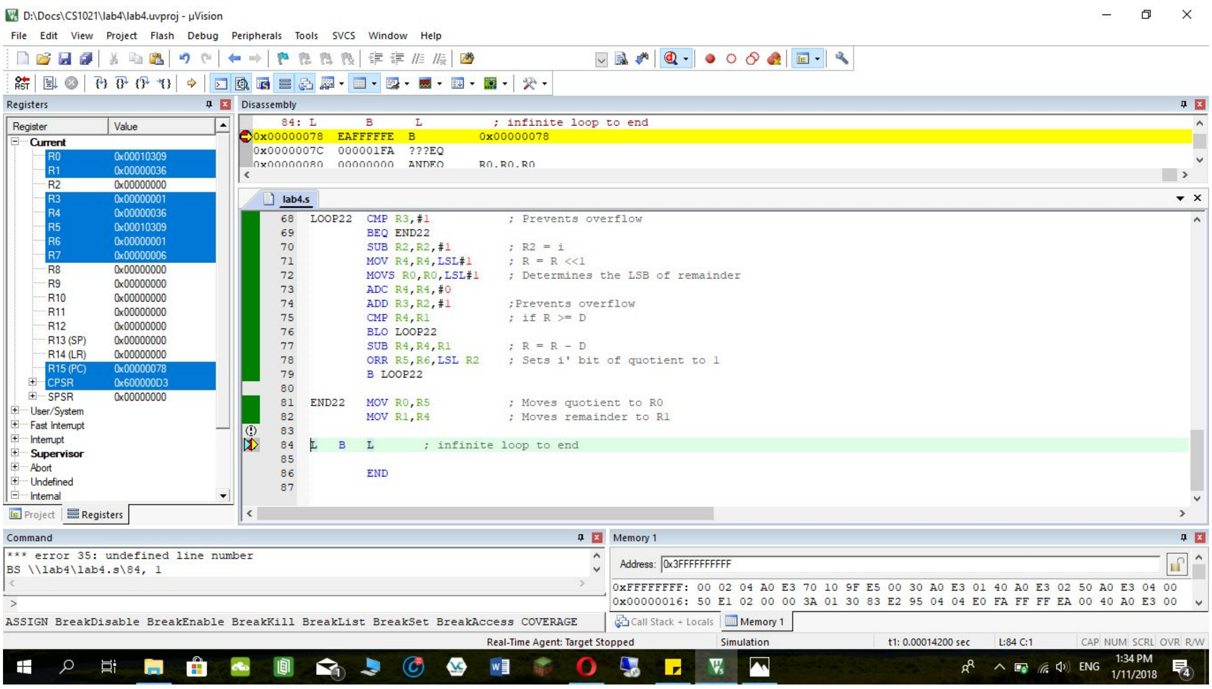
Name: YI XIANG TAN
Student ID:18315028

CS1021 Lab 4 Report



Name: YI XIANG TAN
Student ID:<u>18315028</u>





Dicussion

-Take the number of calculation per seconds, f
the number of calculations per operation, c
and the number of possibilities, n

Name: YI XIANG TAN
Student ID:18315028

The time required would be
t = c*n/f

-Instead of calculating every bit with algorithm 1, algorithm 2 calculates the result from the highest bit.

Consider this case of decimal division:
N = 400
D = 3

For algorithm 1,
R = 400 - 3, Quotient = 1
R' = 397 - 3, Quotient = 2
.
.
.
R = 1 , Quotient = 133
Algorithm 1 would take 133 calculations.

For algorithm 2,
R = 400 - 3(100) , Quotient = 100
R' = 100 - 3(30) , Quotient = 100 + 30
R'' = 10 - 3(3) , Quotient = 100 + 30 + 3
R''' = 1
For the same question, algorithm 2 would take only 3 calculations.

And so we could see that algorithm 2 is more efficient than algorithm 1 as we scale up the differences of numerator and divisor as well as the value of numerator itself.