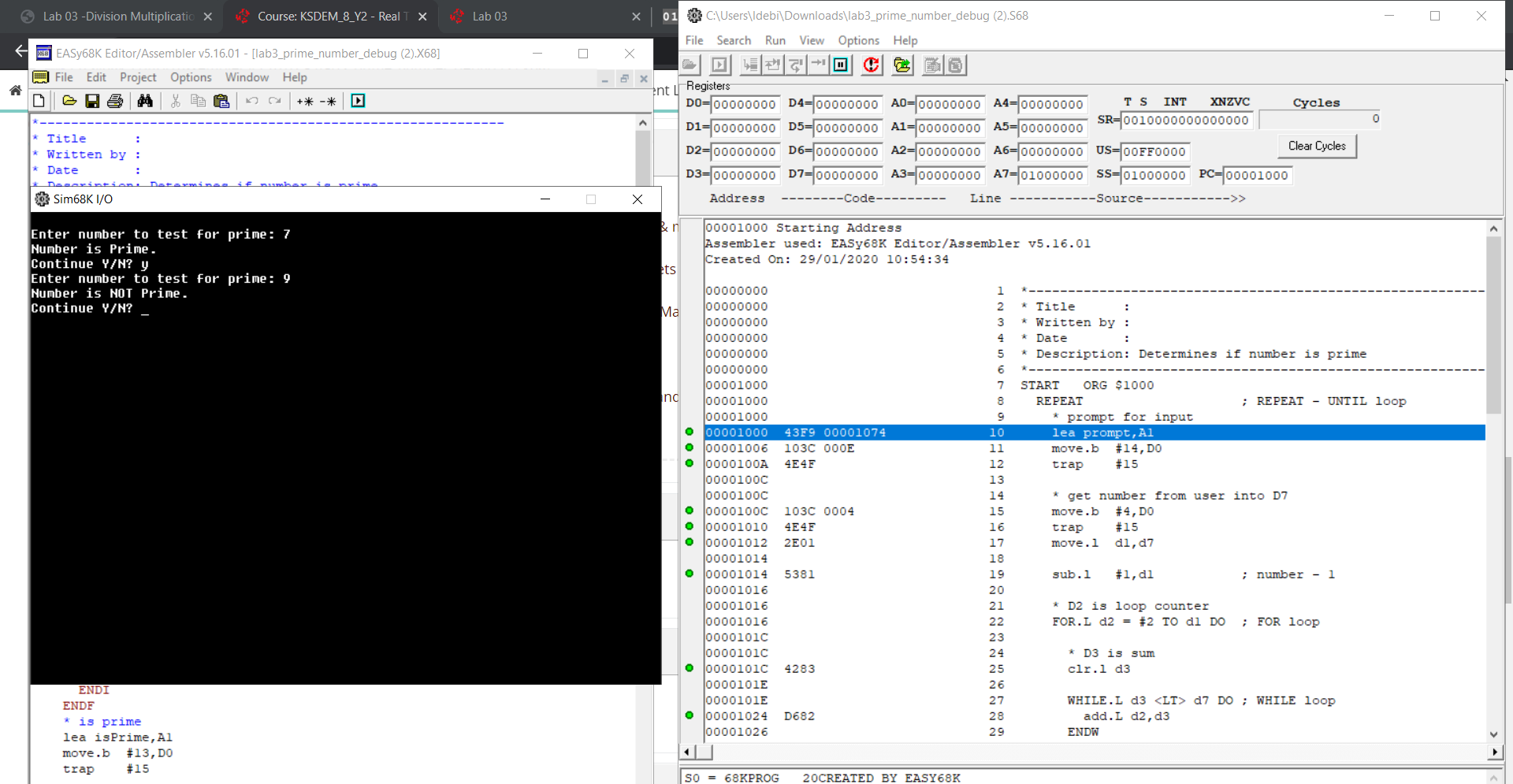
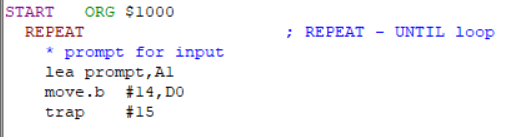
Lab report: Prime numbers debug



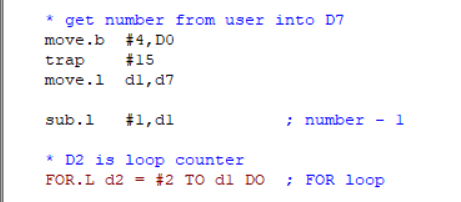
Firstly, we enter an Until loop, that will keep looping until some condition is met further down the code. “LEA prompt, A1” will load the address of the variable labelled prompt into address register A1. “MOVE.B #14,D0” moves the literal value 14 into data register D0, which then TRAP #15 outputs the prompt onto our screen based on D0.



Next, the program gets the number from users input and stores it in D7. It subtracts 1 from D1 (the default register) that it will use to limit the amount of times a for loop loops.

The for loop will keep looping until D2 is equal to D1, and D2 will increment by 1 every time it loops.

In a scenario where the user inputted 7, D1 will be equal to 6 and the loop will repeat 4 times (up to and including 6), the value of D2 increasing by 1 every time.

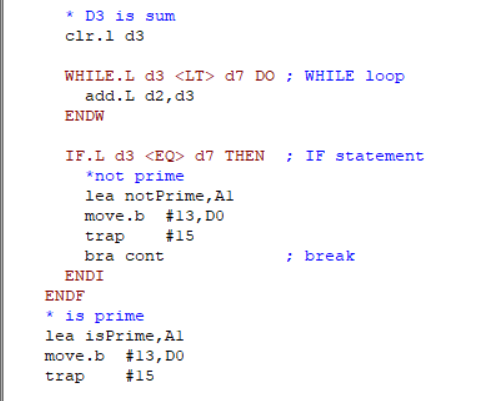


Here we can see that D3 is first cleared. This is done because the program loops and the values from previous loops doesn’t clear by itself. We add D2 to D3 while D3 is less than D7.

D7 is our input, D2 is 2 at the start and D3 is blank. The for loop will keep looping until D2 is equal to D1.

We increase D2 from 2 up to the value of our input -1 (since primes and nonprimes divide by itself). The while loop will break once the value of D3 is equal or bigger than D7.

If D3 is equal to D7, that means it can be divided by another number and it is not prime. We simply output a message that lets the user know that the number isn’t prime and we branch back up to cont. Otherwise, the FOR loop will keep looping D1 times and if a number that can be divided hasn’t been found, then our input number is prime and a message will be outputted.



In a scenario where our input is 7 (PRIME):

|  |  |  |  |
| --- | --- | --- | --- |
| FOR LOOP | D3 | D7 | IF |
| 1 | 8 | 7 | FALSE |
| 2 | 9 | 7 | FALSE |
| 3 | 12 | 7 | FALSE |
| 4 | 10 | 7 | FALSE |

As we can see, the value of D3 does not equal to the value of D7 at any of the 4 time the FOR loop executes, therefore it breaks out of the FOR loop and executes the code underneath, which displays to the user that the number is prime.

In a scenario where our input is 9 (NOT PRIME):

|  |  |  |  |
| --- | --- | --- | --- |
| FOR LOOP | D3 | D7 | IF |
| 1 | 10 | 9 | FALSE |
| 2 | 9 | 9 | TRUE |

In this case, during the second loop where D2 is 3, D3 is equal to 3, then 6 and then 9. Since we specified <LT> in the while loop, the number is equal to D7 and the loop breaks and the IF statement is satisfied, so now the code inside the IF executes and displays to the user that that number isn’t a prime.