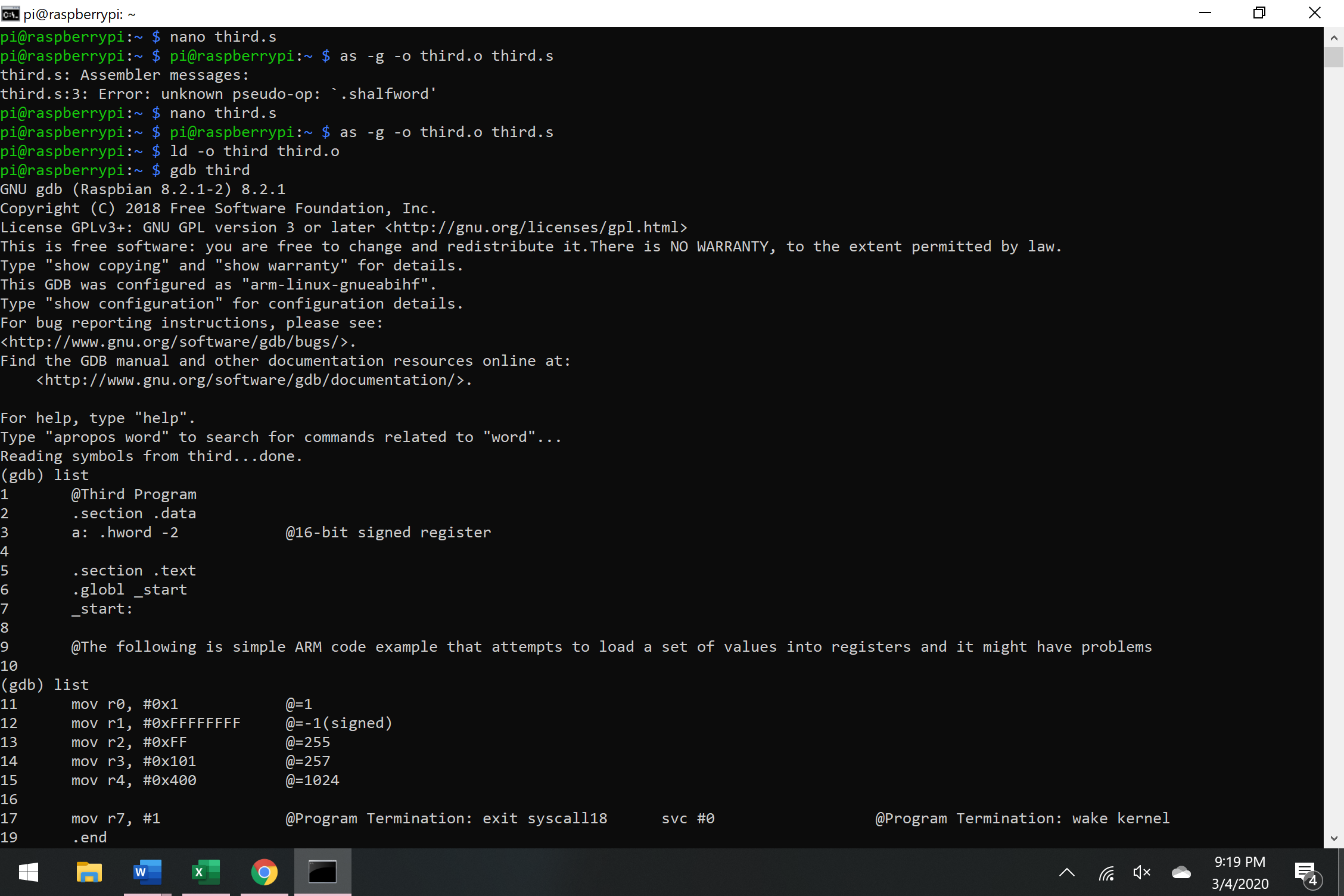
Task 4

I began by typing the given code in Raspberry Pi (Figure 1). After compiling it, I noticed that there was an error in the data section. I noticed that the variable a was wrongly declared as a “shalfword”, therefore I changed it to .hword. Then, I compiled and ran the program using debugger (Figure 2).

A screenshot of a computer screen

Description automatically generated

*Figure 1 – Third program*



*Figure 2 – compiling and running the third program*

I noticed that the values in the registers are just as expected (Figure 3). Register r1 holds a value of 1. Register r2 holds a value of -1, register r3 holds a value of 255 and register r4 holds a value of 257. The values to left are the hexadecimal values of the decimal values we see on the right. The value of -1 is stored as fffffff, which is -1 in two complements.

A screenshot of a cell phone

Description automatically generated

Figure 3 – debugging third program

Part B)

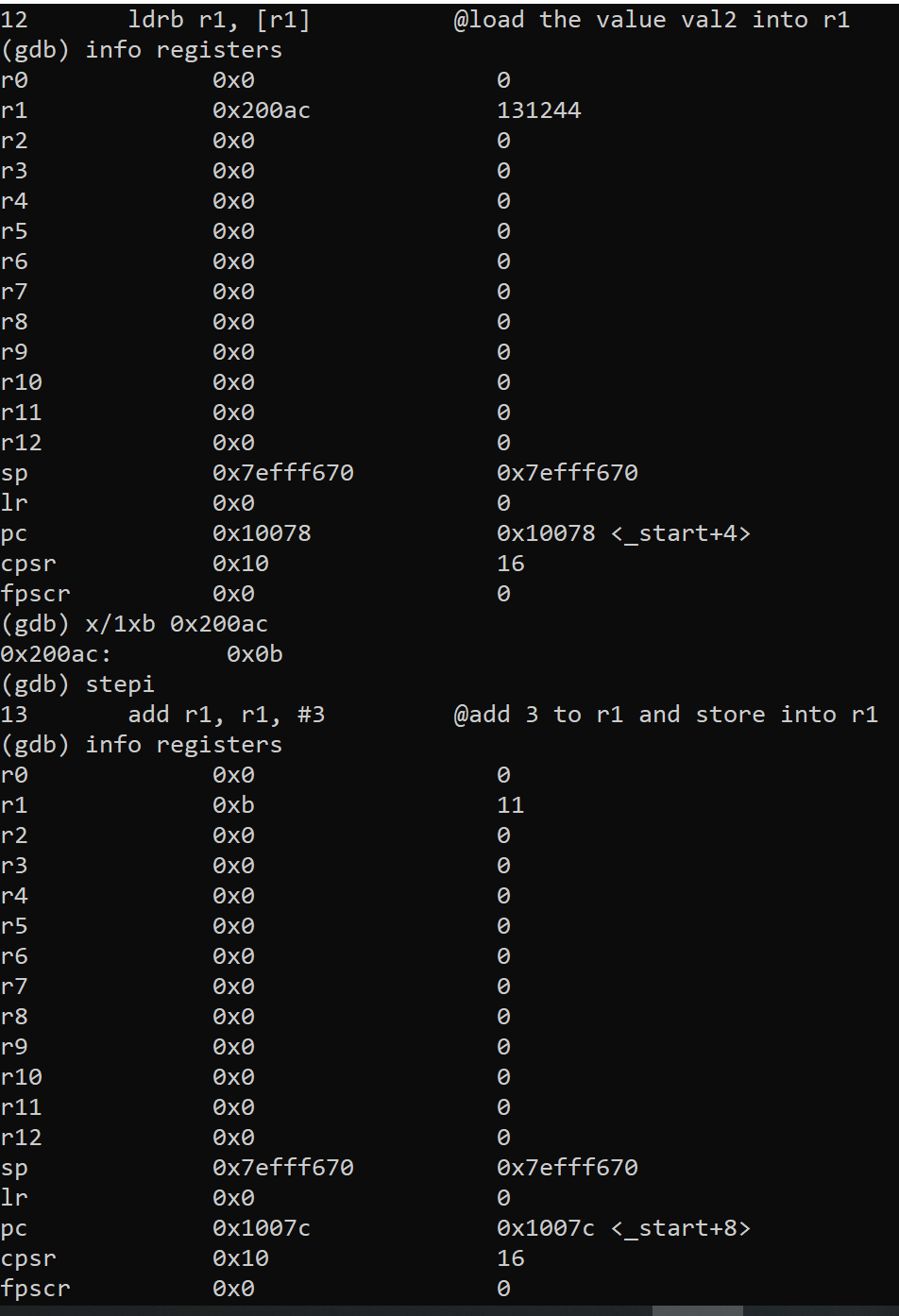
I began by typing the code for arithmetic expression val2+3+val3-val1 (Figure 4).

A screenshot of a cell phone screen with text

Description automatically generated

*Figure 4 – arithmetic3 program*

Then I compiled and debugged the code to check if my code worked just as I expected. First, I loaded val2=11 into register r1 (hexadecimal = b) and check the memory of that register. (Figure 5).



*Figure 5 – debugging arithmetic3*

Then, I added the number 3 to the value on register r1. The register now should show number 14 (hex=e). After that I added the values from r1 and r2 and stored it into register r1. Register r1 after the addition stores the number 30 (hex= 1e) (Figure 6).

A screenshot of a cell phone

Description automatically generated

*Figure 6 – debugging arithmetic3*

Then, I loaded the value of val1 into register r3. Since this is a negative number, to check if the hexadecimal value equals to -60 (hex=c4) I’d have to use two complements (196).Then, in the last arithmetic expression, we are subtracting the val1=-60 from the value from register r1=30. Since our expression equals 30-(-60), the result of that would be 90, which would be stored on register r1 (Figure 7).

However, I notice something interesting happening. The value in register r1 should store the value 90, however in the debugger mode it shows a very large number that equals to hex FFFFFFF5A. This happens because I declared val1=-60 as an unsigned integer, when it is not. Therefore, the register holds a value 0000 0000 C 4. If you want to subtract that number, we first need to use two complements, which will give us the value to 11111111 00111100. This explains why the value that we see is FFFFFFFF5A (Figure 7).

A screenshot of a cell phone

Description automatically generated

*Figure 7 – debugging arithmetic3*

If load instructions for signed integer val1 were used, we could see that the program tuns as it should and the value of r1 is 5A=90 (Figure 8).

*A screenshot of a cell phone

Description automatically generated*

*Figure 8 – the outcome if val1 was declared as signed integer in arithmetic3 program*