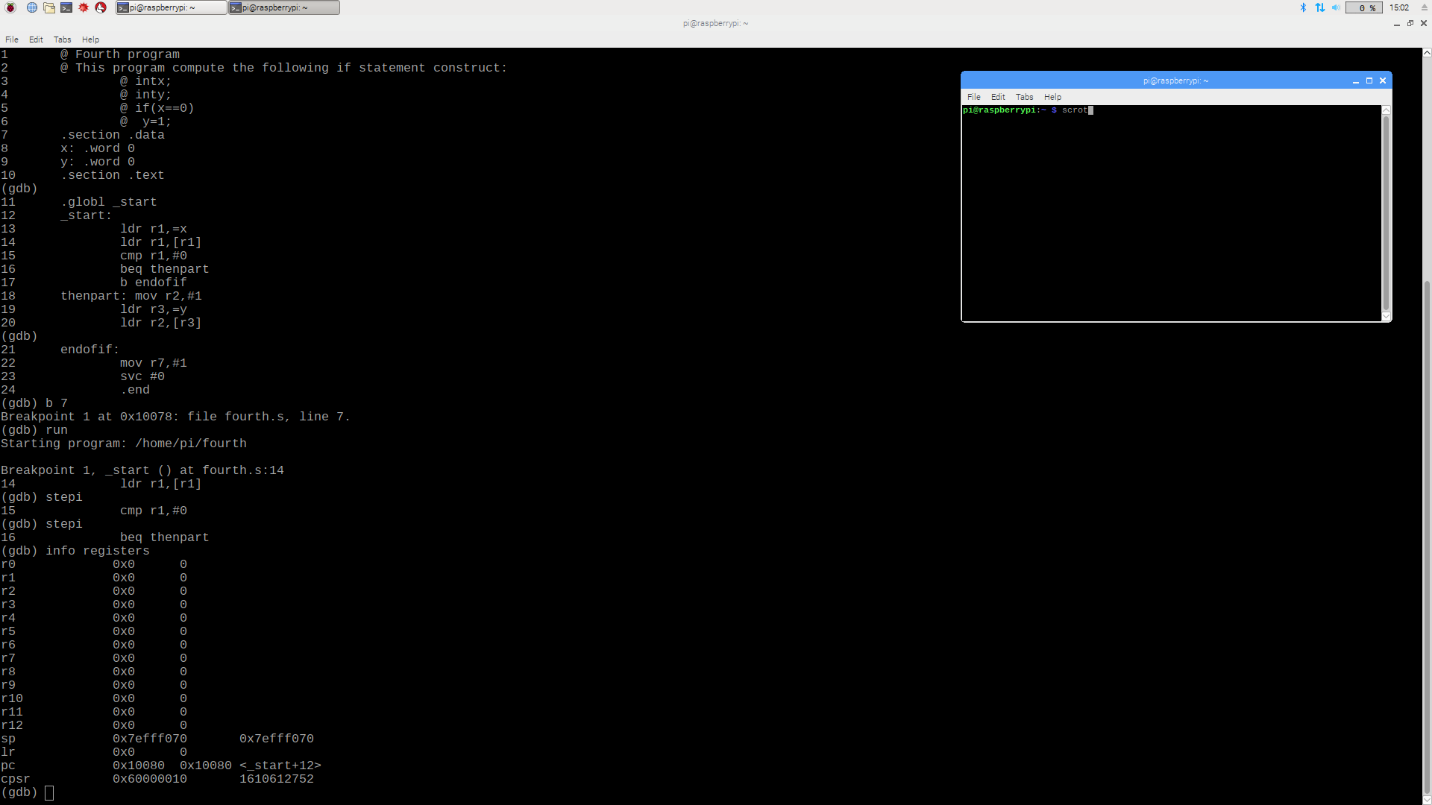
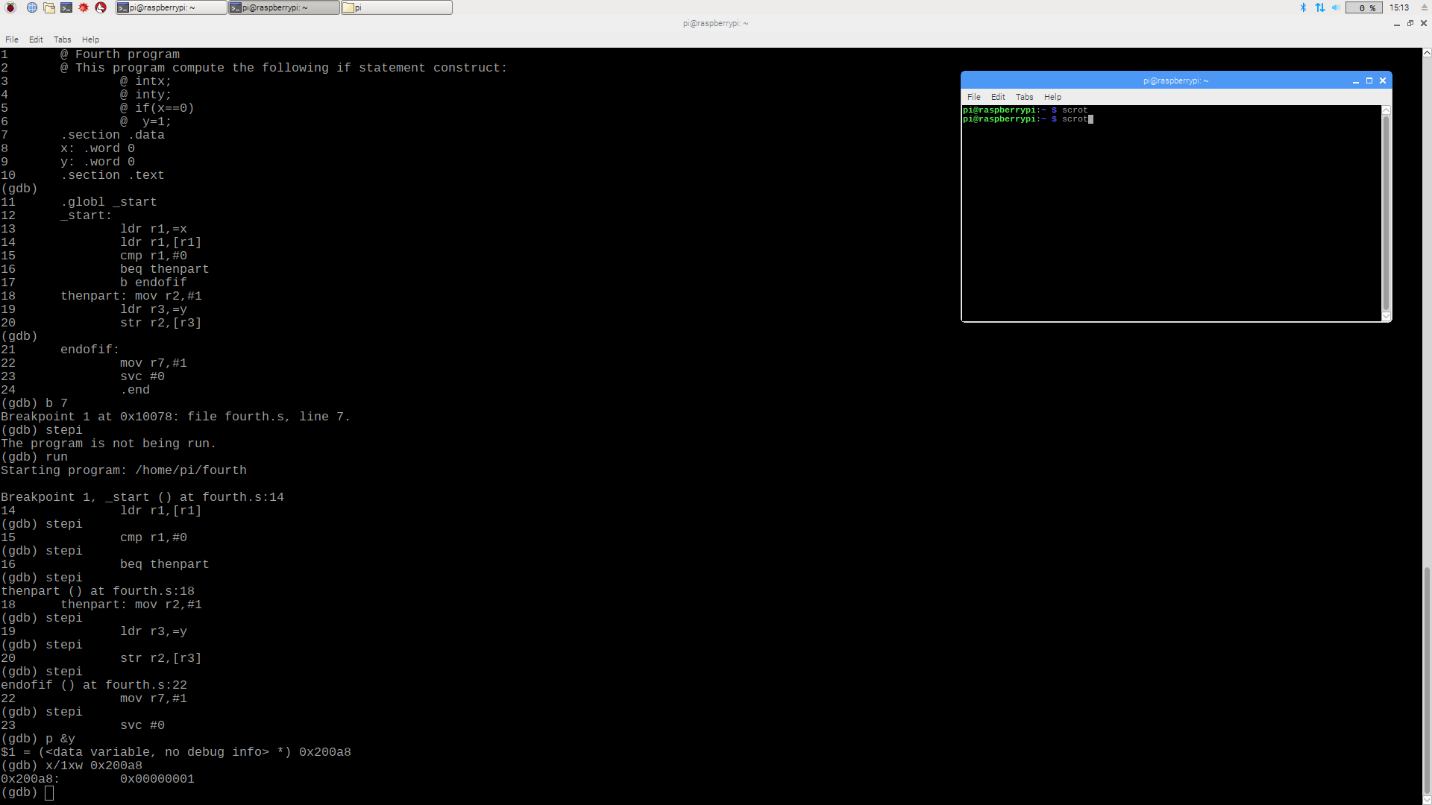
**Arm Assembly Programming Report**

**Part 1:** For part 1 we simply typed out all the code supplied in the instructions. We went through the steps given next in order to assemble and view with gdb. We used ‘p &a’ to find the address of the variable y = 1, then used the x/1xw. First, we put the x and y word value 0 then we start loading into the registers to find if the y = 1. Start from loading x value in r1 then we stored that value in r1, then we compared r1 with 0. If the comparison is true, then it goes to thenpart otherwise its false goes to endofif. Such that the value is true so Z flag is 1 because you can see the register below 0x60000010. 6 in decimal is 0110. After following with the code, we put breakpoint on line 7 then run the entire program by stepi. There was a mistake in assignment on line 20 it supposes to have str instead of ldr because we cannot store two different registers. Hence, we prove the value y =1.

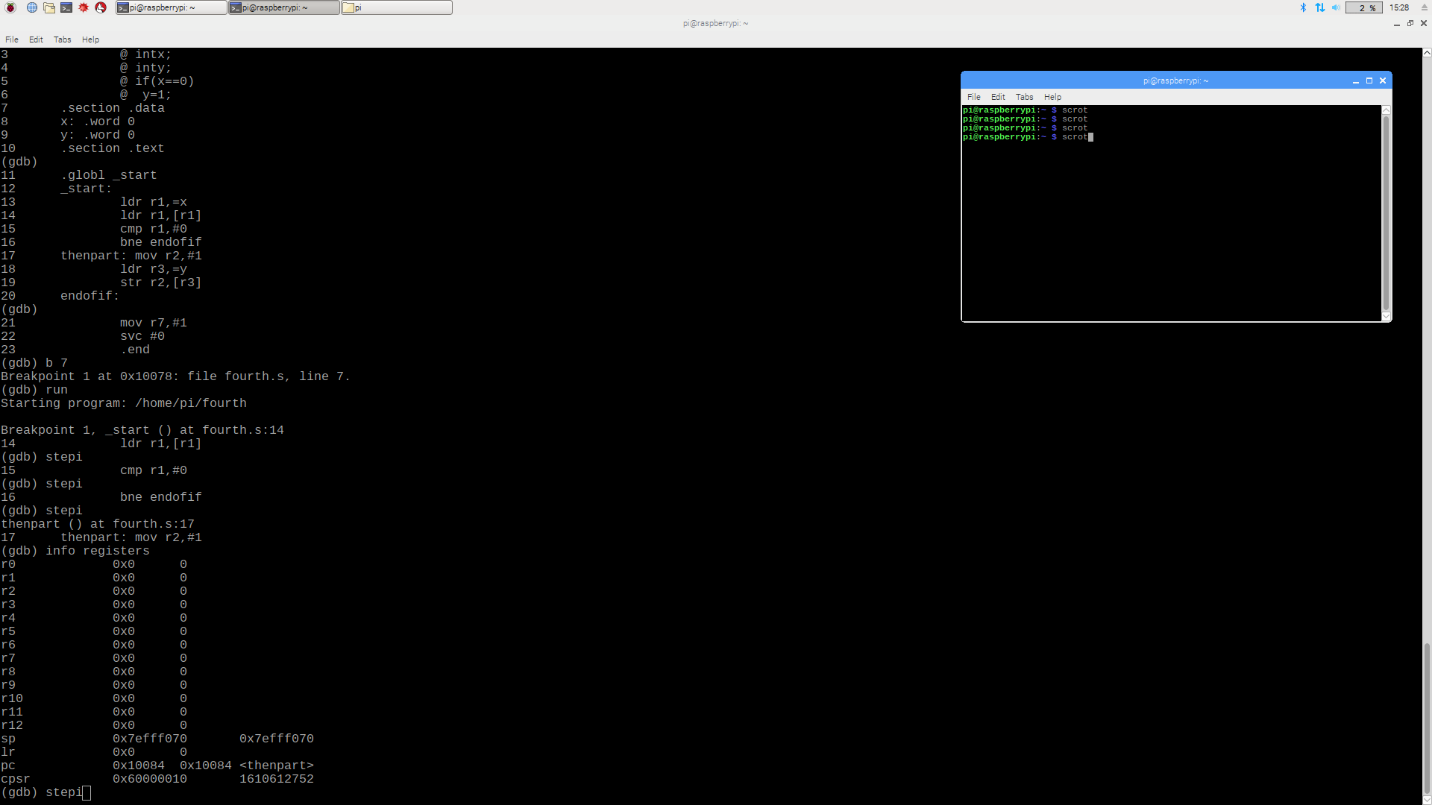


*See Appendix (Arm registers part 1).*

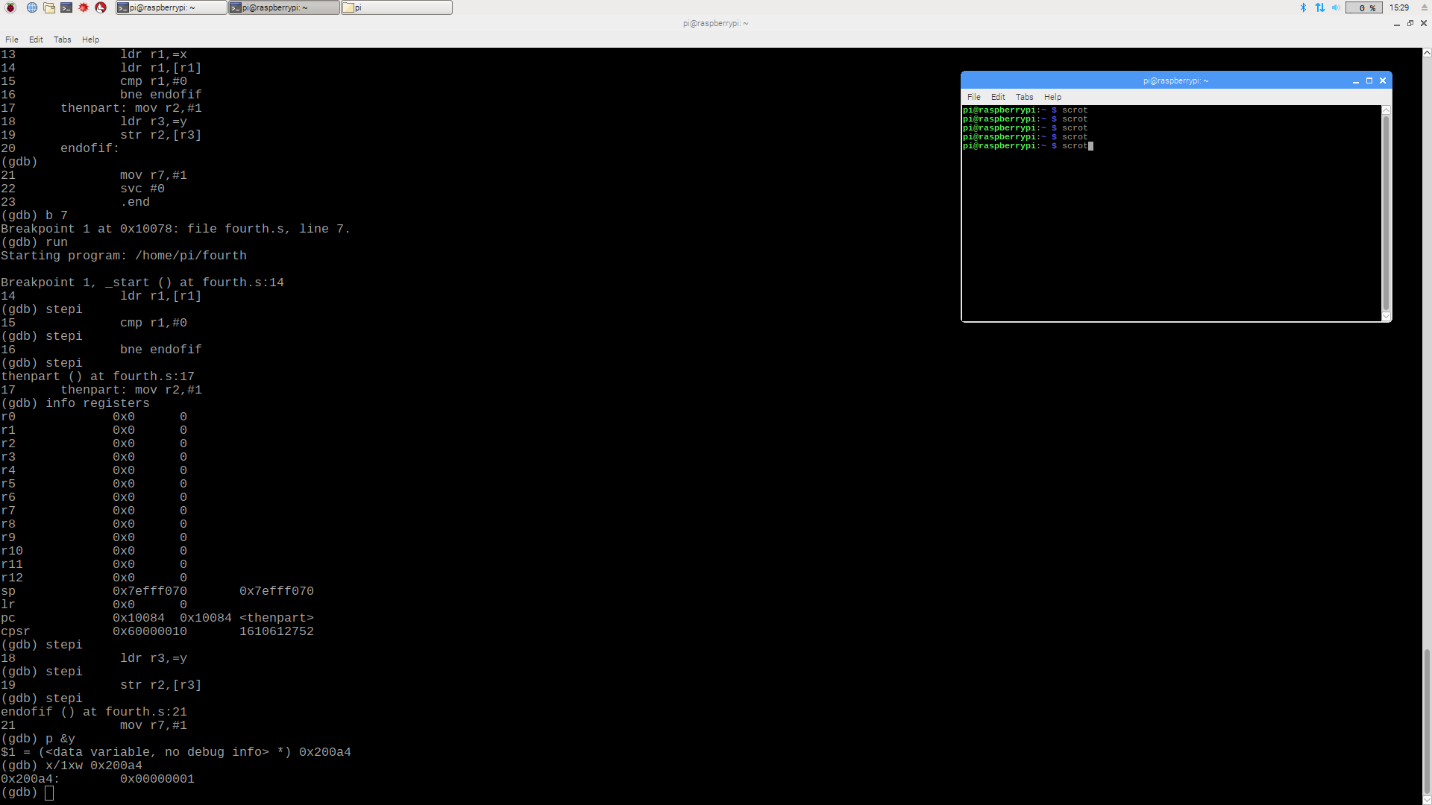


*See Appendix (Arm Y value part 1).*

**Part 2:** For part 2 we remove beq then part and convert the line 16 from b endofif to bne endofif because if branch on not equal (z = 0) then it goes to endofif part but eventually z value is 1 exact same as part 1. Also y value is 1.

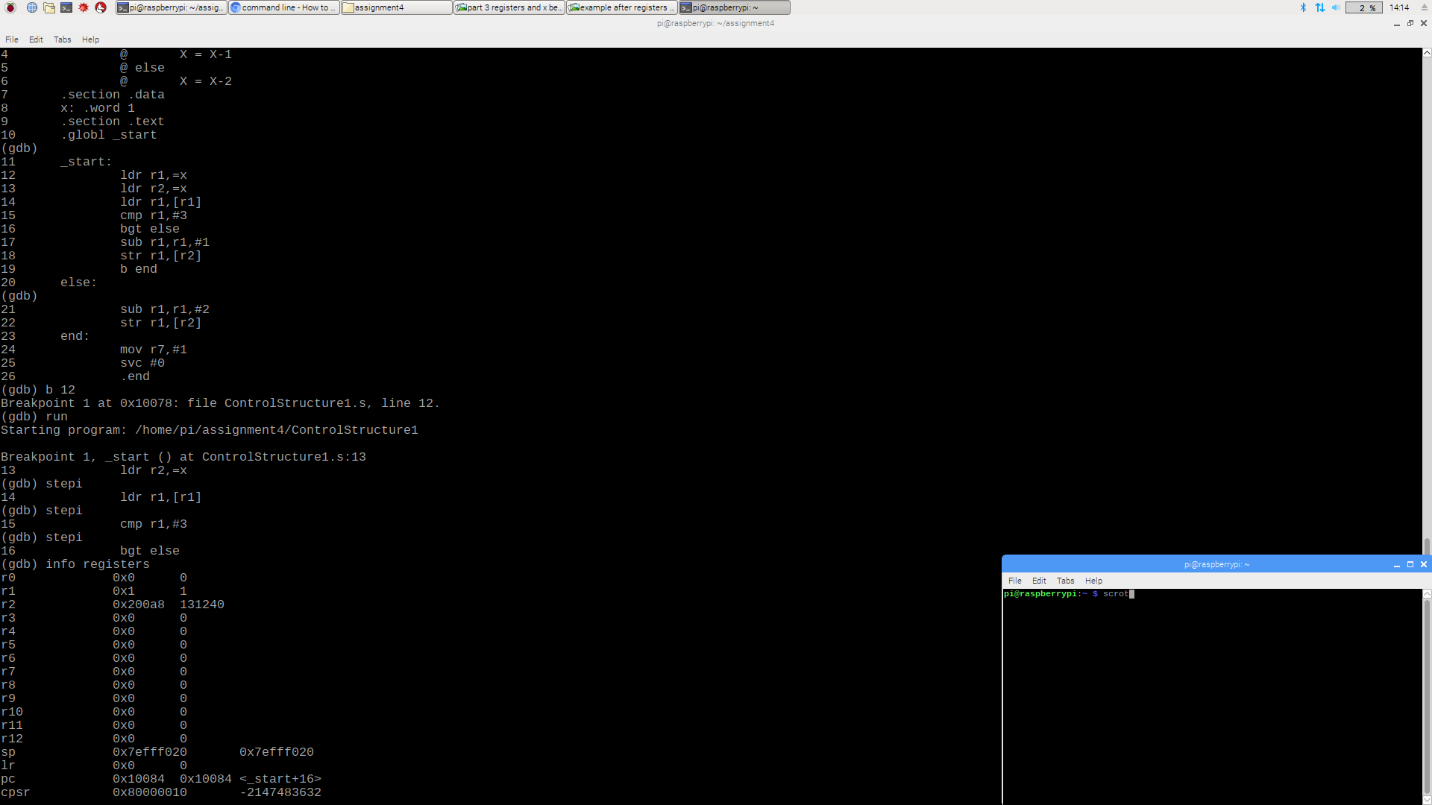


*See Appendix (Arm registers part 2).*

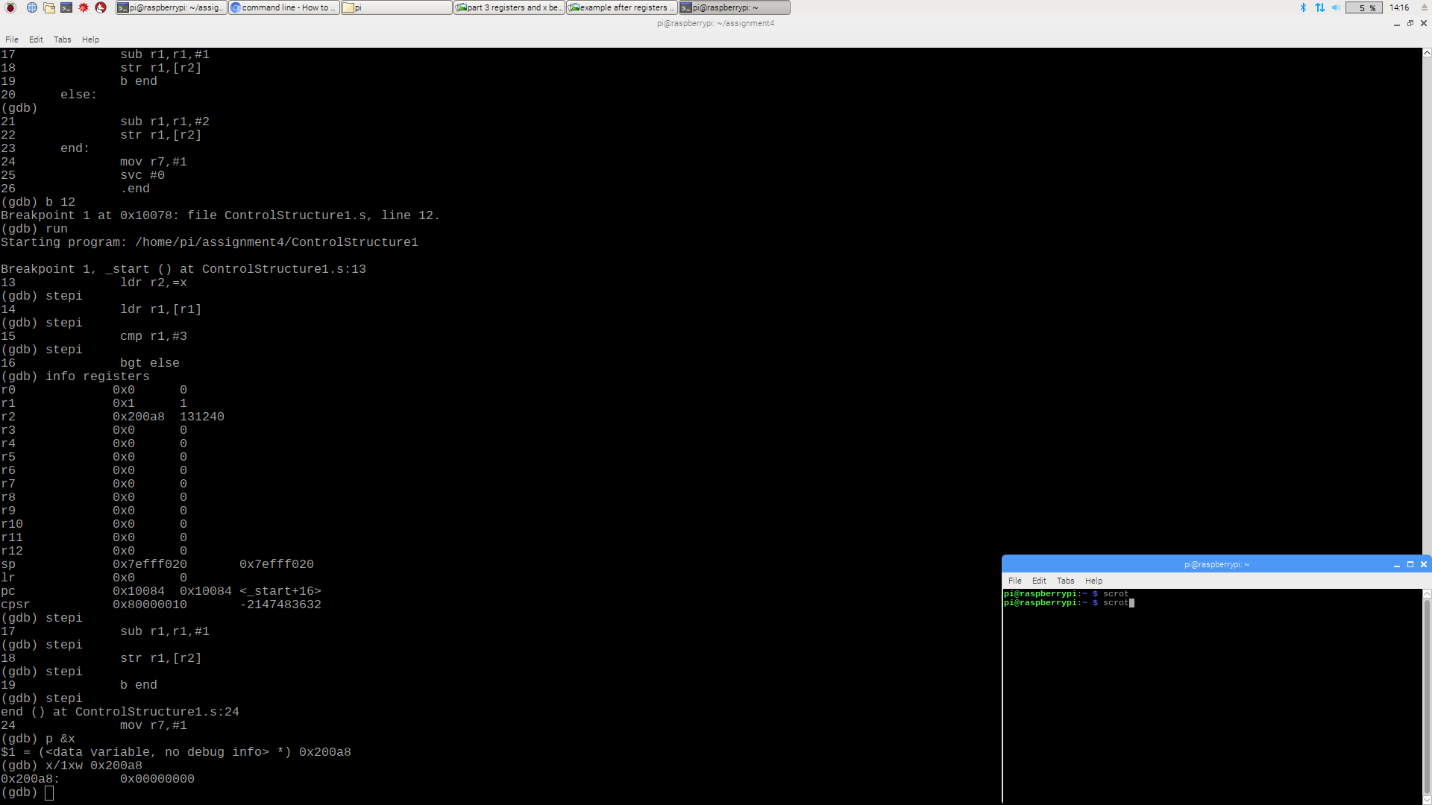


*See Appendix (Arm Y value part 2).*

**Part 3:** For part 3 we changed the entire program and name it ControlStructure1. First, we put the comment that if X less then or equal to 3, X is equal to x – 1 else X is equal to x – 2. We take two registers r1 and r2 to load then store r1 value in r1 and compare with =3. We use bgt means branch on greater than. If the value is greater than 3 it subtracts 1 from r1 and str r2 value in r1 but in this case the value of r1 is not greater than 3 so it goes to else statement and subtract r1 value from 2 and store the r2 value in r1. As you can see the registers cpsr value 0x80000010, so z flag value is 1 because 8 is 1111. After run the program by stepi we put gdb p &x to find the data variable which is 0x200a8 and then gdb x/1xw 0x200a8 to find the x value which is 0.



*See Appendix (Arm registers part 3)*



*See Appendix (Arm x value part 3)*