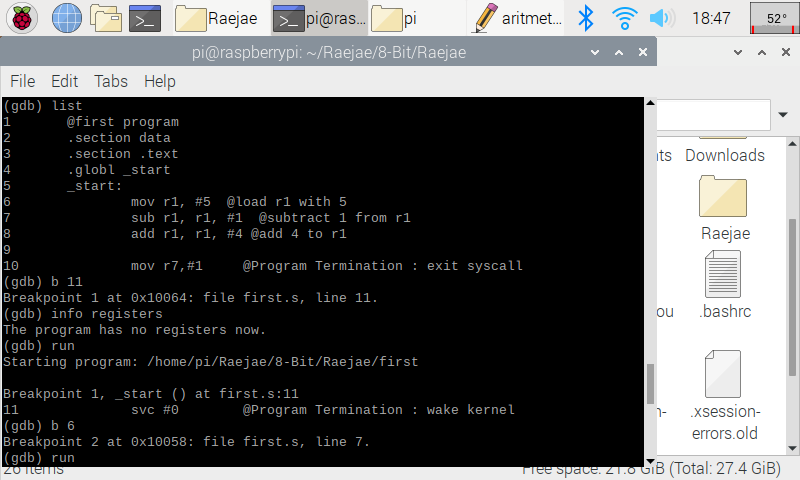
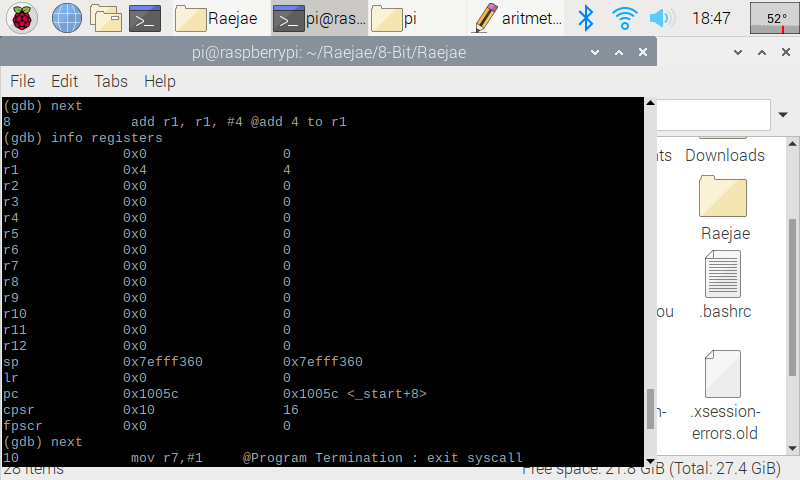
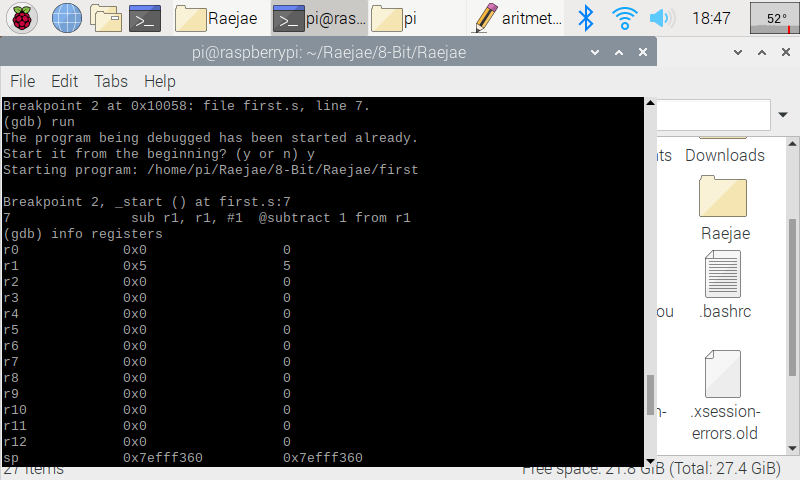


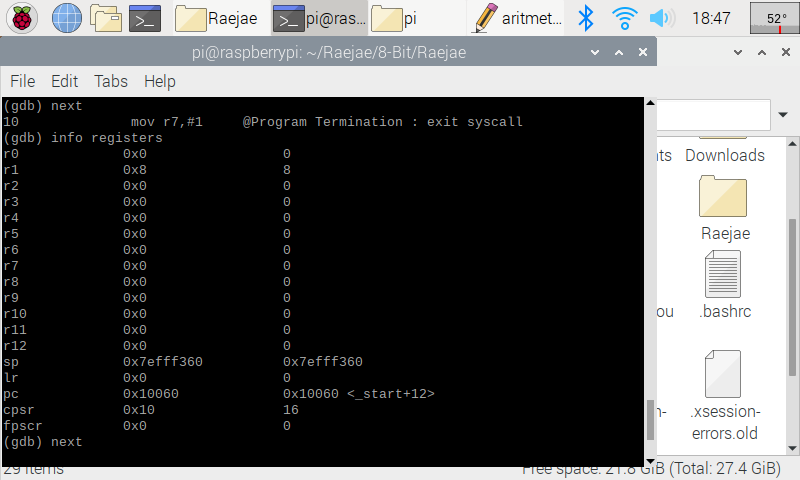
Basic Initialization of code.



This is applying breakpoints. By adding a breakpoint at beginning and end it allows us to continuously run through our code in debugger line by line.

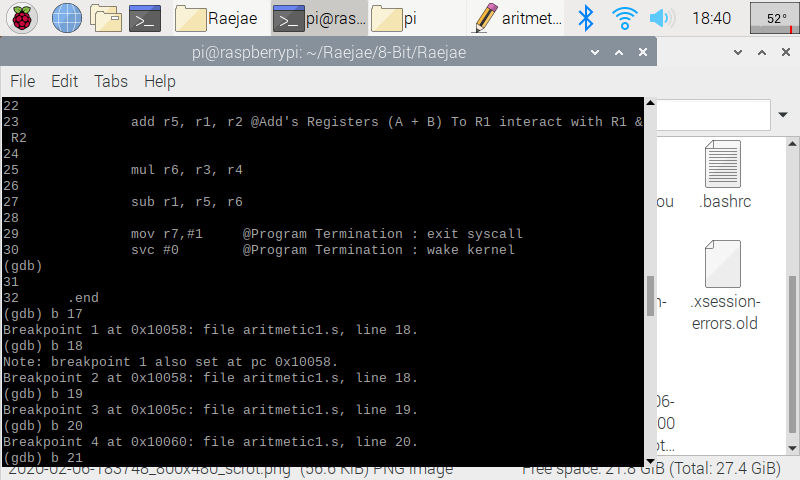
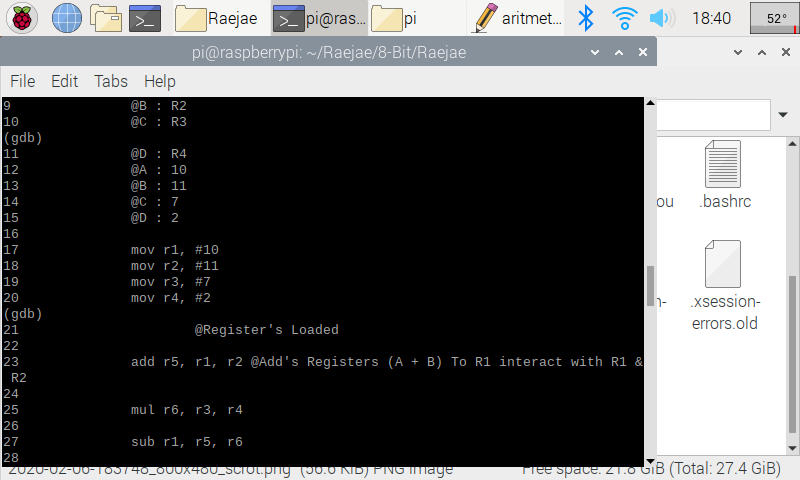


On this line we load 5 to the Register (r1) : and afterwards we subtract 1 from the same register which lands us on 4 for the first register.

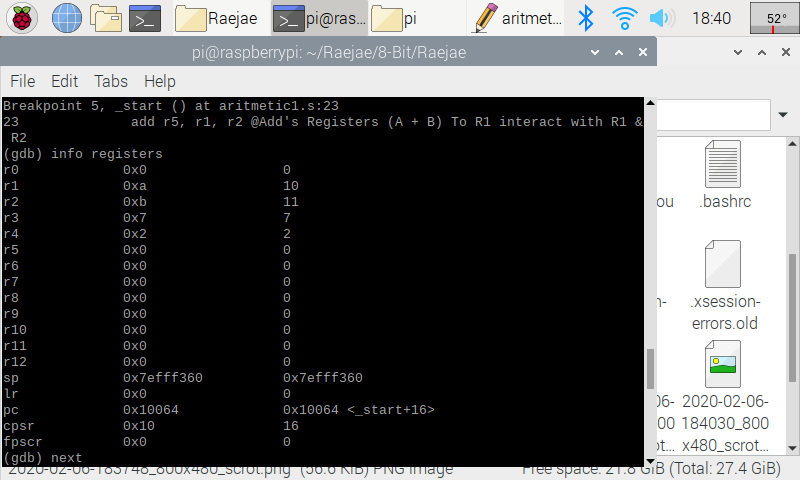




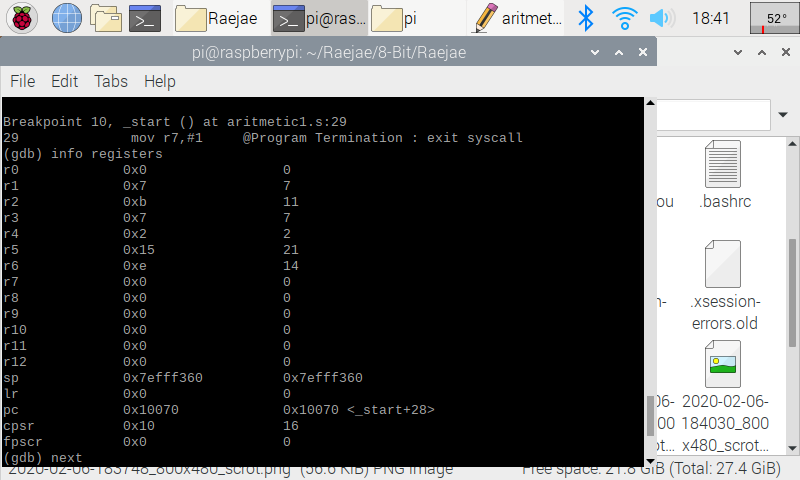
On the lines above we add 4 to the R1 register which updates this register and then we load to R7 #1.



In the two snaps above, after loading in Arithmetic similarly to “First.” After setting breakpoints manually I realized that you could set the initial and end and the program will run through. Unfortunately I realized too late and found myself taking a long route.



The first few lines of code are simply dedicated to moving numbers in the registers but an important thing to notice is that in r1 and r2 there is a 0xa / 0xb which lets us know this is hexadecimal representation.



This is the result after the function calls ADD SUB and MUL which manipulate the registers and updates each answer to a new register below its calling