**Victoria Routon – x87 group**

The first part of the assignment was to create a program in Raspberry PI (Figure1).

A screen shot of a computer

Description automatically generated

Figure 1 – first assembly program

After running the program, I noticed that the program didn’t produce the output. This was to be expected because we did not ask it to print anything (Figure 2).

A black and silver text on a screen

Description automatically generated

Figure 2 – running the first program

Even though the program didn’t have any print statements, I can see how the program runs by using the debugger (Figure 3).A black and silver text on a screen

Description automatically generated

Figure 3 – running the debugger

By typing the command “list” I can see the entire code. This is extremely helpful as it lets me see the code lines, which I can use to set up my debugger’s breakpoints. I chose my breakpoint to be at the line 10 (Figure 4).A close up of a black background

Description automatically generatedFigure 4 – display of the command lines

Then, I typed the command “info registers” which lets me see the output of my code, which is to store value 1 at the register r7 (Figure 5).A screen shot of a computer

Description automatically generated

Figure 5 – registers output

The next part of the project was to create a file named “arithmetic1” (Figure 6).

A screenshot of a computer

Description automatically generated

Figure 6 – arithmetic1 program

Then, just like with the first program, I could not see the output when I ran it. Therefore, I ran the debugger (Figure 7).A black and silver text on a screen

Description automatically generatedFigure 7 – debugging arithmetic1 program

I typed the command “list” to see the command lines (Figure 8).A close up of text on a black background

Description automatically generated

Figure 8 – display of command lines

Since my program had few arithmetic operations, I decided to set up 3 breakpoints – at line 10,12 and 14. After each breakpoints, I displayed the output of the registers. At the line 10 breakpoint, I expect the values A(10),B(11),C(7) and D(2) to be loaded on the registers r1,r2,r3,r4 respectively (Figure 9).

A screenshot of a computer

Description automatically generated

Figure 9 – registers output at line 10

At the line 12, the code executed two arithmetic operations. Adding the values from the register r1 and r2 and loading it on the register r5 (A+B=11). As well as multiplication of the values from the register r3 and r4 and loading it on the register r6 (C\*D=14) (Figure 10).A screen shot of a computer

Description automatically generated

Figure 10 – registers output at line 12

At the line 14, the program subtracted the value of the register r6 (14) from the register r5 (21-14=7) and loaded the result on the register r1 (Figure 11).

A close up of a computer screen

Description automatically generated

Figure 11 –registers output after line 13