NOTE:
some of the instructions differ from some of the knowledge I find online, for example the sub instruction. SUB ra, rb, refers to rb = rb - ra, however the question says its rb = ra-rb. I have went with the assignment descriptions.

Question 3 - Instruction Set 3 – x295++

B. Compiling and assembling a C program using our x295++ instruction set

Table 1

C program (The C code below cannot be modified – it must stay as it is stated below)	**x295++ assembly program**	**x295++ machine code program**
`z = (x + y) * (x - y)`	**LOAD src1, r0** **LOAD src2, r1** **COPY r0, r2** **ADD r1, r2** **SUB r0, r1** **MUL r1, r2** **STORE r2, Dest** **Expand the table as needed!**	1010 000 XXX XXXXXX 0000 <Src1 12 bits> 1010 001 XXX XXXXXX 0000 <Src2 12 bits> 1001 010 000 XXXXXX 0001 010 001 XXXXXX 0010 001 000 XXXXXX 0011 010 001 XXXXXX 1011 XXX 010 XXXXXX 0000 <Dest 12 bits>

C. **Evaluating our x295++ instruction set using Memory Traffic criteria**

Table 2

x295++ program **(1 assembly instruction/ machine code instruction per row)**	**Fetch** **(number of memory accesses of word size)** **+** **Explain how you obtain your count**	**Decode/Execute** **(number of memory accesses of word size)** **+** **Explain how you obtain your count**
Assembly instruction: `LOAD src1, r0` **Machine code instruction:** 1010 000 XXX XXXXXX 0000 <Src1 12 bits>	**Count: 2** **Explanation:** **The instruction 32 bit long, which is twice the size of the word.**	**Count: 1** **Explanation:** **Src1 is the only memory access**
Assembly instruction: `LOAD src2, r1` **Machine code instruction:** 1010 001 XXX XXXXXX 0000 <Src2 12 bits>	**Count: 2** **Explanation:** **The instruction 32 bit long, which is twice the size of the word.**	**Count: 1** **Explanation:** **Src2 is the only memory access**
Assembly instruction: `COPY r0, r2` **Machine code instruction:** 1001 010 000 XXXXXX	**Count: 1** **Explanation: The instruction is 16 bit long, which is 1 word size, thus only 1 memory access needed.**	**Count: 0** **Explanation:** **No memory access needed**
Assembly instruction: `ADD r1, r2` **Machine code instruction:** 0001 010 001 XXXXXX	**Count: 1** **Explanation: The instruction is 16 bit long, which is 1 word size, thus only 1 memory access needed.**	**Count: 0** **Explanation: No memory access needed**

Assembly instruction: `SUB r0, r1` **Machine code instruction:** 0010 001 000 XXXXXX	**Count: 1** **Explanation:** **The instruction is 16 bit long, which is 1 word size, thus only 1 memory access needed.**	**Count: 0** **Explanation:** **No memory access needed**
Assembly instruction: `MUL r1, r2` **Machine code instruction:** 0011 010 001 XXXXXX	**Count: 1** **Explanation:** **The instruction is 16 bit long, which is 1 word size, thus only 1 memory access needed.**	**Count:** **Explanation:** **No memory access needed**
Assembly instruction: `STORE r2, Dest` **Machine code instruction:** 1011 XXX 010 XXXXXX 0000 <Dest 12 bits>	**Count: 2** **Explanation:** **The instruction 32 bit long, which is twice the size of the word.**	**Count: 1** **Explanation:** **Dest is the only memory access**
Add as many rows as you need in order to answer this question!		
Grand Total: 13	**Total: 10**	**Total: 3**

Challenge Question - Instruction Set 3 – x295++

B. Compiling and assembling your modified C program using our x295++ instruction set

Table 1 - Challenge

C program **Modify/simplify the C code below – must still produce the same result as the original C program listed below**	**x295++ assembly program**	**x295++ machine code program**

`z = (x + y) * (x - y)`
 	
`modified/simplified:`
`z = x*x - y*y`
	LOAD src1, r0
	LOAD src2, r1
	MUL r0, r0
	MUL r1, r1
	SUB r0, r1
	STORE r1, Dest
	
	Expand the table as needed!
		1010 000 XXX XXXXXX 0000 <Src1 12 bits>
		1010 001 XXX XXXXXX 0000 <Src2 12 bits>
		0011 000 000 XXXXXX
		0011 001 001 XXXXXX
		0010 001 000 XXXXXX
		1011 XXX 001 XXXXXX 0000 <Dest 12 bits>

C. **Evaluating our x295++ instruction set using Memory Traffic criteria**

Table 2 - Challenge

x295++ program **(1 assembly instruction/ machine code instruction per row)**	**Fetch** **(number of memory accesses of word size)** **+** **Explain how you obtain your count**	**Decode/Execute** **(number of memory accesses of word size)** **+** **Explain how you obtain your count**
Assembly instruction: `LOAD src1, r0` **Machine code instruction:** 1010 000 XXX XXXXXX 0000 <Src1 12 bits>	**Count: 2** **Explanation:** **The instruction 32 bit long, which is twice the size of the word.**	**Count: 1** **Explanation:** **Src1 is the only memory access**
Assembly instruction: `LOAD src2, r1` **Machine code instruction:** 1010 001 XXX XXXXXX 0000 <Src2 12 bits>	**Count: 2** **Explanation:** **The instruction 32 bit long, which is twice the size of the word.**	**Count: 1** **Explanation:** **Src2 is the only memory access**
Assembly instruction: `MUL r0, r0` **Machine code instruction:** 0011 000 000 XXXXXX	**Count: 1** **Explanation:** **The instruction is 16 bit long, which is 1 word size, thus only 1 memory access needed.**	**Count: 0** **Explanation:** **No memory access needed**
Assembly instruction: `MUL r1, r1`	**Count: 1**	**Count: 0**

Machine code instruction: 0011 001 001 XXXXXX	**Explanation:** **The instruction is 16 bit long, which is 1 word size, thus only 1 memory access needed.**	**Explanation:** **No memory access needed**
Assembly instruction: `SUB r0, r1` **Machine code instruction:** 0010 001 000 XXXXXX	**Count: 1** **Explanation:** **The instruction is 16 bit long, which is 1 word size, thus only 1 memory access needed.**	**Count: 0** **Explanation:** **No memory access needed**
Assembly instruction: `STORE r1, Dest` **Machine code instruction:** 1011 XXX 001 XXXXXX 0000 <Dest 12 bits>	**Count: 2** **Explanation:** **The instruction 32 bit long, which is twice the size of the word.**	**Count: 1** **Explanation:** **Dest is the only memory access**
Add as many rows as you need in order to answer this question!		
Grand Total: 12	**Total: 9**	**Total: 3**

Once completed, submit it on Crowdmark as your answer to Question 3.