Question 3 - Instruction Set 3 – x295++ - SOLUTION

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

Table 1

C program	x295++ assembly program	x295++ machine code program
z = (x + y) * (x − y)	LOAD x, r0	1010 000 XXX XXXXXX 0000 <Src 12 bits - x>
	LOAD y, r1	1010 001 XXX XXXXXX 0000 <Src 12 bits - y>
	COPY r1, r2	1001 010 001 XXXXXX
	ADD r0, r1	0001 001 000 XXXXXX
	SUB r0, r2	0010 010 000 XXXXXX
	MUL r1, r2	0011 010 001 XXXXXX
	STORE r2, z	1011 XXX 010 XXXXXX 0000 <Dest 12 bits - z>

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 word size memory accesses)** **+** **Provide an explanation explaining the count**	**Decode/Execute** **(number of word size memory accesses)** **+** **Provide an explanation explaining the count**

Assembly instruction:
`LOAD x, r0`
Machine code instruction:
`1010 000 XXX XXXXXX`
`0000 <Src 12 bits - x>`
	Count: 2
	Explanation: fetching 1 instruction that is 2-word wide
		Count: 1
		Explanation: executing a LOAD requires 1 memory access -> reading value x (16 bits -> 1 word) from memory

Assembly instruction:
`LOAD y, r1`
Machine code instruction:
`1010 001 XXX XXXXXX`
`0000 <Src 12 bits - y>`
	Count: 2
	Explanation: fetching 1 instruction that is 2-word wide
		Count: 1
		Explanation: executing a LOAD requires 1 memory access -> reading value y (16 bits -> 1 word) from memory

Assembly instruction:
`COPY r1, r2`
Machine code instruction:
`1001 010 001 XXXXXX`
	Count: 1
	Explanation: fetching 1 instruction that is 1-word wide
		Count: 0
		Explanation: executing an COPY does not require memory access since it is manipulating values held in registers

Assembly instruction: `ADD r0, r1` **Machine code instruction:**	**Count:** 1 **Explanation:** fetching 1 instruction that is 1-word wide	**Count:** 0 **Explanation:** executing an ADD does not require memory access since it is

`0001 001 000 XXXXXX`		manipulating values held in registers

Assembly instruction:
`SUB r0, r2`
Machine code instruction:
`0010 010 000 XXXXXX`
	Count: 1
	Explanation: fetching 1 instruction that is 1 word wide
		Count: 0
		Explanation: executing an SUB does not require memory access since it is manipulating values held in registers

Assembly instruction:
`MUL r1, r2`
Machine code instruction:
`0011 010 001 XXXXXX`
	Count: 1
	Explanation: fetching 1 instruction that is 1-word wide
		Count: 0
		Explanation: executing an MUL does not require memory access since it is manipulating values held in registers

Assembly instruction:
`STORE r2, z`
Machine code instruction:
`1011 XXX 010 XXXXXX`
`0000 <Dest 12 bits - z>`
	Count: 2
	Explanation: fetching 1 instruction that is 2-word wide
		Count: 1
		Explanation: executing a STORE requires 1 memory access -> writing value z (16 bits -> 1 word) to memory

Grand Total: 13	**Total: 10**	**Total: 3**

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

One more thing …

Challenge: Can you rewrite your **x295++** program above (assembly program and machine code program) such that it has fewer instructions? This would require you to first "re-organize" the above C program.

Note: When rewriting your **x295++** program above using fewer instructions, you do not have to use ALL the assembly instructions defined by our **x295++** instruction set.