As part of your project, answer the following questions. You may discuss the concepts with

others in the class, but each of you must submit your answers in your own words.

1. What opcode will blank memory initialized to 0x00 look like to the processor?

0x00 = 0000 0000

0000 is a memory operation

0 is a store

0 is in the acc

00 means its used as an address and since ACC will be blank the address will be 000

2. Of the 256 possible opcodes we can get from and 8-bit opcode, how many are not being used

in our instruction set, i.e., how many instructions could we add for future expansions of our

Processor?

The math operations have 4 separate sections with 1, 8, 4 and 4 different possibilities. This means the math operations take up 1\*8\*4\*4 = 128

The memory operations have 5 separate sections with 1, 2, 2, and 3, different possinilities. This means the math operations take up 1\*2\*2\*3 = 12

The Branches/jumps have 2 separate sections with 1, and 7 different possinilities. This means the math operations take up 1\*7 = 7

There are two special opcodes 2 = 2

So a total of 128 + 12 + 7 + 2= 149 opcodes are used and 256 – 149 = 107 opcodes are free

3. What would we need to add to our simulator to be able to include the following instructions:

compare ACC with a constant, PUSH to or PULL from the stack, and take the 2's complement of

ACC?

All off these would need an opcode to be assigned to that operation. The ACC compare would need somewhere to temporarily hold the result. Push and Pull would require a regester to hold the head of the stack. The 2s complement can be found using regular math operations and would not need anything new.

4. If executeInstruction() were divided into two parts, decode and execute, what additional

global resources would be needed for your simulator?

You would need to have the operation to be performed and the memory source and desitination store as global variables.