Ashley Nguyen - apn2my
PostLab07 - postlab7.pdf
3/23/16

IBCM requires extreme attention to detail and of course, meticulousness. Regarding the syntax, the linear memory spot feature made editing IBCM code a hassle and inconvenience. In terms of assembly logic, I thoroughly enjoyed writing single stream code while having to keep usually just one value in mind (the accumulator value). Coding loops was initially quite confusing but once I understood how IBCM breaks the loop into multiple chunks and operates based on those chunks through jmp commands. After fully understanding this, IBCM became easier to use but still poses as a challenge when I am testing code out and need to make edits but doing so would alter the memory bits of the opcode, which was frustrating. As for modifications to the simulator, perhaps coloring/highlighting the value box once the value at that spot in memory has updated to convey to the user that a value update has occurred. I feel semi-confident in writing IBCM code. Honestly I need more time to transition from memorizing the general scheme of a for or if loop and visualize the code in terms of the accumulator to confidently go about writing **ANY** program in IBCM.