Problem 3:

Operation code (opcode): specifies the operation to be performed such as add, subtract, shift, branch, etc. An opcode field of k bits can support up to 2^k different operations.

ALU: is a digital circuit that performs arithmetic and logic operations. It takes 2 operands, one comes from AC and other one's from any register that is connected to the data bus, to perform an operation; the result then is put back into the AC.

Effective Address (EA): a k-bit address encoded in an instruction that directly refers to an operand in memory.

Program Counter (PC): is a special register which holds the address of the next instruction to be fetched from memory. Its content is incremented after each instruction execution.

Internal Data Bus: a shared wire that is used to connect various components (memory, registers, ALU, busses, and control unit) in Pseudo-CPU.

Problem 4:

Cycle 1: $TEMP \leftarrow AC$, $MDR \leftarrow M[MAR]$; move content of AC to TEMP and read operand
Cycle 2: $AC \leftarrow MDR$; move operand to AC
Cycle 3: $AC \leftarrow AC + 1$; increment AC
Cycle 4: $MDR \leftarrow AC$; move AC content to MDR
Cycle 5: $M[MAR] \leftarrow MDR$, $AC \leftarrow TEMP$; store new value back into memory and
; restore original value of AC

Problem 5:

Cycle 1: $TEMP \leftarrow AC$, $MDR \leftarrow M[MAR]$; move AC to TEMP and read EA (M(x))
Cycle 2: $AC \leftarrow MDR$; move EA to AC
Cycle 3: $AC \leftarrow AC - 1$; decrement AC ($EA - 1$)
Cycle 4: $MDR \leftarrow AC$; move AC to MDR
Cycle 5: $M[MAR] \leftarrow MDR$, $AC \leftarrow AC + 1$; store AC content back into M(x), increment AC
Cycle 6: $MAR \leftarrow AC$; move EA to MAR
Cycle 7: $AC \leftarrow TEMP$, $MDR \leftarrow TEMP$; restore original AC content, move it to MDR
Cycle 8: $M[MAR] \leftarrow MDR$; store original AC content to memory