Program #1 (see class_cpu.pdf)

H (R1, 5)
Reg: send R1; Memory: write to "a"
Reg: send R1; Display:

Program #2

H(R1,5)
Reg: Send R1; Mem: write to "a"
H (R3, 7)
~~Reg: send R3; Mem: write to~~
A
Reg: Send R1; Mem: write to "a"
Reg: Send R1; Display

Program #3

H(R1,5)
~~Sto~~
Reg: Send R1; Mem: write to "a"
H(R3, 17)
Reg: Send R3; Mem: write to "b"
If 'negative Jump
A
Reg: send R1; Mem: write to "a"
Reg: send R1; Display

Program #4

R3 : ;
R2 : 4
R1 : a
R0 : 1

H(R1, 0)
Reg : send R1; Mem : write to "a"
H(R3, 0)
Reg : send R3; Mem write to ";"
H(R2, 4)
H(R0, 1)

U
IF ! negative Jump
A
Reg : send R1, Mem : write to "a"
A'
Reg : send R3, Mem : write to ";"
Jump]

Reg : send R1 ; Display

Program #5

H(R1, 5)
Reg: Send R1; Mem: write to "a"
H(R3, 100)
C
Reg: R1; Mem: write to "b"
Reg: R1; Display

Note: 500 does not fit in "b". (max value is 127), so the most significant bits are lost.

Program #6
H(R1, 5)
Reg: Send R1; Mem: write to "a"
H(R3, 100)
C
Reg: R0 (send); Mem: write to upper byte of "b"
Reg: Send R1; Mem: write to lower byte of "b"
Reg: send R0; Display
Reg: Send R1; Display

Note: Variable "b" is now 2 bytes. (an "upper" and a "lower byte")

the value (assuming a positive number) is:

U x 256 + L
where U & L are bytes

Program #7

Assume a new operator !

C' R0 x R3 → [R4, R5] multiply

H(R1, 0)
Reg: Send R1; Mem: write to "a"

H(R3, 1)
Reg: Send R3; Mem: write to "j"

H(R0, 2)
H(R2, 0)

R0 : 2
R1 : a
R2 : 0
R3 : 1

U
If zero Jump
E
Reg: Send R1; Mem: write to "a"
Reg: Send R1; ~~Mem~~ Display

C'
J(R3, R5)
Jump 7

multiply by 2
Copy result to R3

(no more code