```
Lui  0x384
Addi R5, R1,20
Xor R3, R1, R5
Lw R1, 0(R0)    ; reg[1] <- mem[0]
Lw R2, 1(R0)    reg[2] <- mem[1]
Lw R3, 2(R0)    reg[3] <- mem[2]
Addi R4, R4, 10
Sub R4, R4, R4
Add R4, R2, R4
Slt R6, R2, R3
Beq R6, R0, 2   If R6=R0 go forward 2 instructions
Add R2, R1, R2
Beq R0, R0, -5  go back 5 instructions
Sw R4, 0(R0)    mem[0] <- reg[4]
Jal func
Sll R3, R2, R5
Add R5, R5, R5
Func: or R5, R2, R3
Lw R1, 0(R0)
Lw R2, 5(R1)    reg[2] <- mem[60]
```

```
Lw R3 ,6(R1)    reg[3] <- mem[61]
And R4, R2, R3
Sw R4, 0(R0)    mem[0] <- reg[4]
Jr R7
```

Memory starting from address 0 contains:

M[0]= 0001

M[1]=0001

M[2]=000a

M[60]=430a

M[61]=7342