Problem 1: [18 points] Drill problem
Filename: hw8prob1a.asm
AndrewID: xinyew

```
; count the position of highest bit of 1 in R6
; store the result in R1
        .ORG    $100
start   LI      R6, $FF ; load R6
        LI      R1, $0  ; set R1 to 0 for init

loop    SLTI    R0, R6, $0  ; test R6 ?= 0
        BRZ     done        ; stop if yes
        SRLI    R6, R6, $1  ; logic right shift R6 by 1
        ADDI    R1, R1, $1  ; add 1 to R1
        BRA     loop        ; continue

done    STOP
```

Problem 1: [18 points] Drill problem
Filename: hw8prob1b.asm
AndrewID: xinyew

```
; calculate R6 * R2
; store the result in R1
        .ORG    $100
start   LI      R6, $FF ; load R6
        LI      R2, $3  ; set R2 to 3 for init
        LI      R1, $0  ; set R1 to 0 for init
        LI      R3, $1  ;

loop    SLTI    R0, R2, $0  ; test R2 ?= 0
        BRZ     done        ; stop if yes
        SUB     R2, R2, R3  ; subtract 1 from R2
        ADD     R1, R1, R6  ; add R1 = R1 + R6
        BRA     loop        ; continue

done    STOP
```

Problem 1: [18 points] Drill problem
Filename: hw8prob1c.asm
AndrewID: xinyew

```
; calculate R4 + (R4-1) + (R4-2) + ... + 1 + 0
; store the result in R1
        .ORG    $100
start   LI      R4, $FF ; load R6
        LI      R1, $0  ; set R1 to 0 for init
        LI      R3, $1  ; set R3 to 1 for subtraction

loop    SLTI    R0, R4, $0  ; test R4 ?= 0
        BRZ     done        ; stop if yes
        SUB     R4, R4, R3  ; subtract 1 from R4
        ADD     R1, R1, R4  ; add R1 = R1 + R4
        BRA     loop        ; continue

done    STOP
```

Problem 3: [6 points]
Filename: hw8prob3.asm
AndrewID: xinyew

```
        .ORG    $500
START   LI      r7, $5300
        LI      r5, $5310
        MV      r4, r0
LOOP    LW      r2, r7, $0
        BRN     IF
        BRA     NEXT
IF      SUB     r2, r0, r2
NEXT    ADD     r4, r4, r2
        ADDI    r7, r7, $2
        SLT     r0, r7, r5
        BRNZ    LOOP
FINISH  SW      r7, r4, $0
        STOP

        .ORG    $5300
        .DW     $0000
        .DW     $FFFF
        .DW     $0002
        .DW     $FFFD
        .DW     $0004
        .DW     $FFFB
        .DW     $0006
        .DW     $FFF9
        .DW     $0008
        .DW     $FFF7
```

Problem 4: [12 points]
Filename: hw8prob4.asm
AndrewID: xinyew

```
        .ORG    $0300
START   LW      r1, r0, $2000   ; r1 = len(array), for counting loops
        LI      r2, $FFF8       ; r2 = -8, for
        LI      r3, $0000       ; r3 = 0, for indexing the array
        LI      r6, $FFFF       ; r6 = -1, for deducting from r1
        LI      r7, $4          ; r7 = -4, for getting the 3rd LSB

LOOP    SLT     r0, r0, r1
        BRZ     DONE            ; if r1 == 0, done
        LW      r4, r3, $2002   ; r4 = M[r3 + 2002], tmp var storing values
        LI      r7, $4          ; r7 = 4, for testing whether to round
        AND     r5, r4, r7      ; r5 = r4 & r7, tmp var storing camparison
        BRZ     MODIFY          ; if the 3rd LSB is 0, do the branching
        ADDI    r4, r4, $8      ; otherwise add 8 to the array element

MODIFY  AND     r4, r4, r2      ; check all bits other than the 3 LSBs
        BRN     NEG             ; if negative, branching to NEG round processing
        BRA     STORE           ; if positive, branching to STORE directly

NEG     SUB     r4, r0, r4      ; negate the value

STORE   SW      r3, r4, $4000   ; store the modified number to memory
        ADD     r1, r1, r6      ; loop counter -= 1
        ADDI    r3, r3, $2      ; index += 2
        BRA     loop            ; continue

DONE    STOP

        .ORG    $2000
        .DW     $5
        .DW     $0001
        .DW     $0101
        .DW     $0005
        .DW     $0105
        .DW     $8001
```