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
```

```
 1 ; calculate R6 * R2
 2 ; store the result in R1
 3         .ORG    $100
 4 start   LI      R6, $FF ; load R6
 5         LI      R2, $3  ; set R2 to 3 for init
 6         LI      R1, $0  ; set R1 to 0 for init
 7         LI      R3, $1  ;
 8
 9 loop    SLTI    R0, R2, $0  ; test R2 ?= 0
10         BRZ     done        ; stop if yes
11         SUB     R2, R2, R3  ; subtract 1 from R2
12         ADD     R1, R1, R6  ; add R1 = R1 + R6
13         BRA     loop        ; continue
14
15 done    STOP
```

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

```
 1 ; calculate R4 + (R4-1) + (R4-2) + ... + 1 + 0
 2 ; store the result in R1
 3         .ORG    $100
 4 start   LI      R4, $FF ; load R6
 5         LI      R1, $0  ; set R1 to 0 for init
 6         LI      R3, $1  ; set R3 to 1 for subtraction
 7
 8 loop    SLTI    R0, R4, $0  ; test R4 ?= 0
 9         BRZ     done        ; stop if yes
10         SUB     R4, R4, R3  ; subtract 1 from R4
11         ADD     R1, R1, R4  ; add R1 = R1 + R4
12         BRA     loop        ; continue
13
14 done    STOP
```

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

```
 1          .ORG    $500
 2 START    LI      r7, $5300
 3          LI      r5, $5310
 4          MV      r4, r0
 5 LOOP     LW      r2, r7, $0
 6          BRN     IF
 7          BRA     NEXT
 8 IF       SUB     r2, r0, r2
 9 NEXT     ADD     r4, r4, r2
10          ADDI    r7, r7, $2
11          SLT     r0, r7, r5
12          BRNZ    LOOP
13 FINISH   SW      r7, r4, $0
14          STOP
15
16          .ORG    $5300
17          .DW     $0000
18          .DW     $FFFF
19          .DW     $0002
20          .DW     $FFFD
21          .DW     $0004
22          .DW     $FFFB
23          .DW     $0006
24          .DW     $FFF9
25          .DW     $0008
26          .DW     $FFF7
```

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

```
 1          .ORG    $0300
 2 START    LW      r1, r0, $2000   ; r1 = len(array), for counting loops
 3          LI      r2, $FFF8       ; r2 = -8, for getting first 13 bits
 4          LI      r3, $0000       ; r3 = 0, for indexing the array
 5          LI      r6, $FFFF       ; r6 = -1, for deducting from r1
 6          LI      r7, $4          ; r7 = 4, for getting the 3rd LSB
 7 LOOP     SLT     r0, r0, r1
 8          BRZ     DONE            ; if r1 == 0, done
 9          LW      r4, r3, $2002   ; r4 = M[r3 + 2002], tmp var storing values
10          AND     r5, r4, r7      ; r5 = r4 & r7, tmp var storing camparison
11          BRZ     MODIFY          ; if the 3rd LSB is 0, do the branching
12          ADDI    r4, r4, $8      ; otherwise add 8 to the array element
13 MODIFY   AND     r4, r4, r2      ; check all bits other than the 3 LSBs
14          BRN     NEG             ; if negative, branching to NEG round processing
15          BRA     STORE           ; if positive, branching to STORE directly
16 NEG      SUB     r4, r0, r4      ; negate the value
17 STORE    SW      r3, r4, $4000   ; store the modified number to memory
18          ADD     r1, r1, r6      ; loop counter -= 1
19          ADDI    r3, r3, $2      ; index += 2
20          BRA     loop            ; continue
21 DONE     STOP
22
23          .ORG    $2000
24          .DW     $5
25          .DW     $0001
26          .DW     $0101
27          .DW     $0005
28          .DW     $0105
29          .DW     $8001
```