Problem 1: [6 points] Drill problem
Filename: hw9prob1a.asm

```
 1 ; puts 15 in r1, 14 in r2, right shifts r2 by 1, puts r1 - r2 in r3
 2  .ORG   $100
 3  LI     R1, $F  ; R1 <- $15 (immediate)
 4  LI     R2, $E   ; R2 <- $14 (immediate)
 5  SRLI   R2, R2, $1 ; R2 <- R2 << 1 (immediate)
 6  SUB    R3, R1, R2 ; R3 <- R1 - R2 (register)
 7  STOP
 8
 9
10
```

Problem 1: [6 points] Drill problem
Filename: hw9prob1b.asm

```
 1 ; sets r6 to 0x23 if r5 < r7; else sets its to 0xBB
 2    .ORG $0
 3    LI R5, $15
 4    LI R7, $17
 5    SLT R0, R5, R7
 6    BRNZ dn
 7 up LI R6, $23
 8    BRA fn
 9 dn LI R6, $BB
10 fn STOP
```

Problem 1: [6 points] Drill problem
Filename: hw9prob1c.asm

```
1  ; Moves $11 around from R4 -> R2 -> R6 in that order, clearing each one after
2  .ORG $0
3  LI R4, $11
4  MV R2, R4
5  MV R4, R0
6  MV R6, R2
7  MV R2, R0
8  STOP
9
```

Problem 3: [12 points] Drill problem
Filename: hw9prob3.asm

```
 1      .ORG $1234
 2 A    .DW $e ; val of a
 3 B    .DW $e ; val of b
 4 RES  .DW $0
 5
 6      .ORG $1000
 7       LW R2, R0, A ; load value of A into R2
 8       LW R3, R0, B ; load value of B into R3
 9       SLTI R0, R2, $0 ; see if A is 0
10       BRZ resa ; go to end, store A in RES if yes
11       SLTI R0, R3, $0 ; see if B is 0
12       BRZ resb ; go to end, store B in RES if yes
13       ; start of the loop
14 loop SLT R4, R2, R3 ; compare the registers
15       BRZ resa    ; equal, store either one in RES
16       BRN altb; a < b
17       ; a > b, so a = a - b
18       SUB R2, R2, R3
19       BRA loop ; back again
20       ; a < b, so b = b - a
21 altb SUB R3, R3, R2
22       BRA loop ; back to loop
23 resa SW R0, R2, RES ; store M[RES] = a
24       BRA done ; go to end
25 resb SW R0, R3, RES ; store M[RES] = b
26 done LW R7, R0, RES;
27       STOP
28
29
30
31
32
33
```

Problem 4: [14 points] Drill problem
Filename: hw9prob4.asm

```
 1
 2          .ORG $2000
 3 BOARD    .DW  $4273;.DW  $1324 ; placements of queens in row 0-3;.
 4          .DW  $6051;DW  $0657 ; placement of queens in row 4-7
 5
 6          .ORG $1000
 7          LI R7, $0 ; initialize board 1 or board2 counter
 8          LI R6, $0 ; initialize the index counter, which is +4 each time
 9          LI R1, $0 ; initialize the how much to shift by counter, mod 12
10          LI R3, $0 ; initialize R3 as our diag counter
11          LI R5, $0 ; initialize R5 as other diag counter
12          LI R2, $0 ; initialize R2 as column counter
13
14 ztothr   LI R7, $0 ; initialize to use board
15          MV R1, R6 ; initializes R1 to R6
16          BRA loop
17
18 frtosev LI R7, $2 ; initial to use board + 1
19          LI R1, $10 ; subtract 16 get ready
20          SUB R1, R6, R1 ; subtract 16 from R6
21
22 loop     LW R4, R7, BOARD ; loads in the word
23          LI R7, $F000 ; initialize the thing to get our bits out
24          SLL R4, R4, R1 ;shift the appropriate amount to the left
25          AND R4, R4, R7 ; get rid of unnecessary bits
26          SRLI R4, R4, $C ; shift over to have bits in rightmost spot, R4 = col
27          SRLI R6, R6, $2 ; Store the row, which is r6 / 4, R6 = row
28
29          ADD R7, R4, R6 ; current value of row + col
30          LI R1, $1 ; get ready to find shift location with R3
31          SLL R1, R1, R7 ; shift by row + col
32          AND R0, R3, R1 ; if 0, its a new TR to BL diag
33          BRZ chklrd ; check TL->BR diag
34          BRA seen
35
36 chklrd   OR R3, R3, R1 ; was ok so set that bit to a 1 in R3
37
38          ADDI R7, R4, $7 ; get R7 = 7 + col
39          SUB R7, R7, R6 ; get R7 = 7 + col - row for TL to BR diag
40          LI R1, $1 ; get ready to find shift location with R5
41          SLL R1, R1, R7 ; R1 shifted by R7 to the left
42          AND R0, R5, R1 ; check if visited; if 0, seen before
43          BRZ chkcol ; check column
44          BRA seen
45
46 chkcol   OR R5, R5, R1 ; was ok so set that bit to a 1 in R5
47          LI R1, $1 ; get ready to find column with R2
48          SLL R1, R1, R4 ; shift by number of columns
49          SLLI R6, R6, $2 ; restore R6 back to a counter
50          AND R0, R2, R1 ; if 0, it's a new column
51          BRZ next
52          BRA seen
53
54 next     OR R2, R2, R1 ; set columns right
55          SLTI R1, R6, $1C ; check if we are at 28 yet
56          BRZ good
57          ADDI R6, R6, $4 ; add 4 to R6
58          SLTI R1, R6, $10 ; check if less than or equal to 20
59          BRN ztothr ; branch if we are 0 - 12
60          BRA frtosev ; branch to 4-7 row
61
62 good     LI R7, $1 ; we are good!
63          BRA done ; go to done
64
65 seen     LI R7, $0 ; already seen before, go home sam
66 done     STOP
67
68
69
70
```

```
 71
 72
 73
 74
```