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

```

; this file loads number 1-5
        .ORG $100
        LI R1, $001      ;
        LI R2, $002      ;
        LI R3, $003      ;
        LI R3, $004      ;
        LI R3, $005      ;
DONE    STOP
```

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

```

; this file coverts the number 21 to its 2's complement form
        .ORG $100
        LI R1, $0015     ;
        NOT  R1,R1       ;
        ADDI  R1,R1,$1   ;
DONE    STOP
```

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

```

; this file has the value 21 and 19 in R1 and R2 then
; R6 adds 19+3 then the value 22 is added to 21
        .ORG $100
        LI R1, $0015     ;
        LI R2, $0014     ;
        ADDI R6,R2,$003  ;
        ADD  R3,R1,R2    ;
DONE    STOP
```

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

```


;   Compute a 9-sided magic square, using RISC240 assembly language
            .ORG   $FF0                     ; Input Data
SIDE        .DW $9                          ; Size of the Magic Square
SQUARE      .DW $51                         ; SIDE * SIDE
BASE        .EQU $2000                      ; Base address of destination array

            .ORG $1000                      ; Code segment

            LI    R3, $51                   ; loads 81 into rs3
            LI    R2, $0;                   ; load 0 into rs1

LOOP        SLLI   R4, R2, $1               ; multiplies i by 2
            SW     R4, R0, BASE             ; m[base + i*2] = 0
            ADDI   R2, R2, $1               ; increments i by 1 or i=i+1
            SLT    R7, R2, R3               ; checks if i < 81
            BRNZ   LOOP                     ; continue the loop
            BRA    ROWSTART                 ; goes to rowstart


;   assigning row and col initially

ROWSTART   LI     R5, SIDE                     ;
           SRAI   R5, R5, $0001                ; col = SIDE // 2 stored in R5
           LI     R6, SIDE                     ;
           SRAI   R6, R6, $0001                ;
           ADDI   R6, R6, $0001                ; row = SIDE // 2 + 1 stored in R6

BEGLOOP    LI     R2, $1                    ; R2 will store i
           LI     R7, SIDE                  ; R7 is a temporary variable
           SLL    R7, R7, R6                ;
           ADD    R7, R7, R5                ;
           SLLI   R7, R7, $0001             ;
           LW     R7, R7, BASE              ;

           MV     R3, R5                    ;
           ADDI   R3, R3, $0001             ; defines nextcol = col + 1 at r3
           SLTI   R7, R3, SIDE              ;
           BRNZ   NEXTVAL1                  ; skips sub if failed
           LI     R7, SIDE                  ; checks if col >= nextcol
           SUB    R3,R3,R7                  ; col = col-1

NEXTVAL1   MV     R4, R6                    ;
           ADDI   R4, R4, $0001             ; defines nextrow = row + 1 at r4
           SLTI   R7, R4, SIDE              ;
           BRNZ   NEXTVAL2                  ;
           LI     R7, SIDE                  ; checks if row >= nextrow
           SUB    R4, R4, R7                ; row = row-1


NEXTVAL2   LI     R7, SIDE                  ;
           SLL    R7, R7, R4                ;
           ADD    R7, R7, R3                ;
           SLLI   R7, R7, $0001             ;
           ADDI   R7, R7, BASE              ;
           LW     R7, R7, BASE              ;
           MV     R1, R2                    ; check if m[addr] == 0
           BRNZ    EXECUTEHERE              ;
           MV     R5, R3                    ; col = nextcol
           MV     R6, R4                    ; row = nextrow

EXECUTEHERE ADDI R6,R6, $2                  ;
           SLTI   R7, R4, SIDE              ;
           BRNZ   NEXTVAL3                  ;
           LI     R7,SIDE                   ;
           SUB    R4, R4, R7                ;

NEXTVAL3    LI   R7, $52                    ;
```

```
            ADDI R2, R2, $1            ;
            SLTI R7, R4, SIDE          ;
            BRN   DONE                 ;
            BRA   BEGLOOP              ;
DONE        STOP;
```