Homework #3

Illya Starikov

Due Date: September 22, 2016

Problem #1

(a) Illegal (*`ADD` only applies to A.*)

(b) Illegal (*`255H` is too large*)

(c) Legal

(d) Illegal (*Same reasoning as (a)*)

(e) Legal

(f) Illegal (*`F5H` will be treated as a label instead of a number.*)

Problem #2

```
MOV R4, #25H      ; Moves 37 into register 4.
MOV A, #1FH       ; Moves 31 into temporary register A
ADD A, R4         ; Adds register 4's value to register A's value
```

It's a simple add operation, resulting in the value of $37_{10} + 31_{10} = 68_{10} = 44_{16}$; it's kept in ROM.

Problem #3

```
    ORG 0;
    MOV R4, #0FH     ; Move 0F to register 4
    MOV A, #0A2H     ; Move A2 to register A
    ADD A, R4        ; Add R4 To A
    MOV R7, A        ; Move the result to R7
    END;
```

Address	Contents
0000	0F
0001	A2
0002	B1
0003	B1

Problem #4

```
        ORG 0

        MOV R0, #30H         ; Move the value 30H to R0
LOOP:   MOV @R0, #0FFH       ; Move FFH at R0
        INC R0               ; Increment R0
        CJNE R0, #40H, LOOP

        END
```

Problem #5

```
    ORG 0;

    SETB PSW.3       ; Switch to register bank 3
    MOV R0, #0A5H    ; Load A5 into register 0
    PUSH 0           ; Push R0 onto stack

    SETB PSW.0       ; Switch to register bank 4
    POP 5            ; Pop onto register 5

    END;
```

Problem #6

Assuming A is the register to store results,

```
    ORG 0;
```

```
MOV R6, #5FH         ; Load 5F into R6
MOV R5, #01001001B   ; Load 01001001 into R5

/* Swap */
MOV A, R6;           ; temp (A) = R6
MOV 6, 5;            ; R6 = R5
MOV R5, A            ; R5 = Temp (A)

/* Add */
ADD A, R6;           ; A already had R6's values, just add original R5
                     ; Which got switched.. so actually add R6
ADD A, #20;

END;
```