

Homework #4

Illya Starikov

Due Date: October 20th, 2016

Problem #1

```
ORG 0

GREEN BIT P2.0
YELLOW BIT P2.1
RED BIT P2.3

MOV P1, #0FFH ; Make an import port

INCT5:
    CPL RED
    ACALL DELAY5S
    CPL RED
    ACALL DELAY5S
    MOV R0, #5H ; Attempt counter
    MOV R2, #30H ; First correct numbers
    MOV R4, 3 ; For yellow light

    SETB C

INCT:
    MOV R1, #3H ; There are three numbers to get right
    JC SKIP ; Basically, if first time, skip the delay

    MOV A, R4
    JZ SKIP1 ; If 3 attempts have been used, yellow
    MOV R4, #0FFH ; To make sure you don't hit this twice
    CPL YELLOW
    ACALL DELAY5S
    CPL YELLOW
```

```

        SJMP SKIP2            ; Because we don't want two delays

SKIP1:  ACALL DELAY5S
SKIP2:  DEC R4

        MOV A, R0
        JZ INCT5              ; A = R0 = 0, then 5 attempts have been made

SKIP:   CLR C
        CRCT:
        MOV 3, @R1            ; R3 = @R1
        MOV A, P1             ; A = P1
        CJNE A, 1, INCT

        INC R2
        DJNZ R1, CRCT;
        SETB P0.0             ; If this far, it's correct

        CPL GREEN
        ACALL DELAY5S
        CPL GREEN
        CLR P0.1              ; Re-lock after five seconds

END

```

Problem #2

```

ORG 0

        SETB PSW.4
        SETB PSW.3           ; set to register bank 3

        MOV A, 7              ; A = B3,R7

        CJNE A, #64, NEXT    ; This is literally to set the flag, nothing else
NEXT:

        CLR PSW.4
        CLR PSW.3           ; set register bank to bank 0

```

```

/*
    R6 IS UPPER BYTE
    R5 IS LOWER BYTE
*/

```

```

MOV B, #100D
DIV AB

```

```

MOV R6, A
MOV R5, B

```

```

END

```

Problem #3

```

ORG 0
MOV R7, #0           ; 1's occurrence counter
MOV R0, #8           ; Loop counter
MOV R1, #30H         ; For writing to ROM
MOV R2, #0           ; Port counter, i.e. will be 0 when P0, 1 when P1

CLR C                ; Just in case

MOV P1, #0FFH        ; Make P1 an input port
MOV A, P1

/* Determine the location of 1s */
ILOOP: RRC A          ; Start scanning at LSB
      INC R2
      JNC SKIP        ; If it is 1, next immediate line will execute
      INC R7          ; Another occurrence of 1
      MOV @R1, 2      ; Rom[30H + i] = Port count
      INC R2
SKIP:  DJNZ R0, ILOOP

/*Output to P2 */
MOV P2, #0           ; Make P2 an output port
REPEAT:              ; For repeating output indefinitely
MOV R1, #30H         ; Again, to be used for output
MOV 3, 7             ; R3 = R7 (The number of occurrences)

```

```

        OLOOP:                ; For outputting a single time
        MOV P2, @R1;          ; P2 = @R1, the value of where the 1 occurred
        ACALL DELAY;
        INC R1;

        DJNZ R3, OLOOP;
        SJMP REPEAT;

DELAY:  MOV R7, #118           ; DJNZ = 2MC. (256/1.085069444) / 2 MC = 117.9648 MC
DLOOP:  DJNZ R7, DLOOP
        RET

        END

```

Problem #4

```

        ORG 0

        RESTART:
        SETB P1.1
        SETB P1.2
        SETB P1.3

        ACALL DELAY           ; 12.5%
        CLR P1.1

        ACALL DELAY           ; 25 %
        CLR P1.2

        ACALL DELAY           ; 37.5%
        ACALL DELAY           ; 50%
        CLR P1.3

        ACALL DELAY           ; 62.5
        ACALL DELAY           ; 75
        ACALL DELAY           ; 87.5
        ACALL DELAY           ; 100

        AJMP RESTART;

DELAY:  MOV R0, 5              ; 1khz = 100 us. So 12.5% = 12.5 us.

```

```
DELL:  DJNZ R0, DELL      ; So ~5 iterations
      RET

      END
```