EEE316 MICROPROCESSORS
PRE-LABORATORY REPORT

NAME : **TURHAN CAN KARGIN**
ID NUMBER : **150403005**
LAB. NUMBER : **3**

OBJECTIVES OF THE LABORATORY ASSIGNMENT:

Objectives of this lab are examining I/O port operations, improving math operations skills and learning Rotate commands.

CODE AND COMMENTS:

1.

```
org 0h
     BANKSEL ANSELC ;initialization steps
     CLRF ANSELC
     BANKSEL ANSELB
     CLRF ANSELB
     BANKSEL ANSELA
     CLRF ANSELA
     CLRF TRISC ;make PORTC output
     CLRF TRISB ;make PORTB output
     SETF TRISA ;set PORTA as input
COUNT1   EQU 0X25 ; COUNT1 = 25H
    ; FIRST PART
Start    movlw B'00000000' ; WREG = 00000000
     movwf PORTB    ; SET PORTB '00000000'
     movwf PORTC    ; SET PORTB '00000000'
     CALL DELAY ; DELAY
     BTFSC PORTA,0  ; checking portc RA0 pin
     BRA AGAIN      ; if it is SET go to AGAIN, otherwise continue
     BTFSC PORTA,1  ; checking portc RA1 pin
     BRA AGAIN2     ; if it is SET go to AGAIN2, otherwise continue
     BTFSC PORTA,2  ; checking portc RA2 pin
     BRA AGAIN3     ; if it is SET go to AGAIN3, otherwise continue
     MOVLW .8 ; we have 8 leds and bits for port
     MOVWF COUNT1
LOP1     RRCF PORTB ; For rotating 8 times from left to right
     BSF STATUS, C ; C=1
     RLCF PORTC ; For rotating 8 times from right to left
     BSF STATUS, C ; C=1
     CALL DELAY
     DECF COUNT1 ; setting 8 times
     BNZ LOP1
     MOVLW .8
     MOVWF COUNT1
```

```
LOP2     BCF STATUS, C ;C=0
     RRCF PORTB ;rotate PORTB 8 times to right
     BCF STATUS, C ;C=0
     RLCF PORTC ;rotate PORTC 8 times to LEFT
     BCF STATUS, C ;C=0
     CALL DELAY
     DECF COUNT1
     BNZ LOP2
     MOVLW .8
     MOVWF COUNT1
LOP3     BSF STATUS, C ;C=1
     RLCF PORTB ;rotate portB 8 times to LEFT
     BSF STATUS, C ;C=1
     RRCF PORTC ;rotate portC 8 times to RIGHT
     BSF STATUS, C ;C=1
     CALL DELAY
     DECF COUNT1
     BNZ LOP3
     MOVLW .8
     MOVWF COUNT1
LOP4     BCF STATUS, C ;C=1
     RLCF PORTB ;rotate portB 8 times to LEFT
     BCF STATUS, C ;C=1
     RRCF PORTC ;rotate portC 8 times to RIGHT
     BCF STATUS, C ;C=1
     CALL DELAY
     DECF COUNT1
     BNZ LOP4
     GOTO Start ; go to begining of this part
    ; SECOND PART
AGAIN    BTFSS PORTA,0   ; checking portc RA0 pin
     BRA Start       ; if it is RESET go to START, otherwise continue
     movlw B'00000001' ;set PORTC 00000001
     movwf PORTC    ;light it with delay
     CALL DELAY
     MOVLW .7
     MOVWF COUNT1
LOP5     RLNCF PORTC ; rotate portb 7 times to left
     CALL DELAY  ; with delay
     DECF COUNT1
     BNZ LOP5
     CLRF PORTC ; PORTC is 00000000
     CALL DELAY
     MOVLW B'00000001' ; start lighting portb with delay
     MOVWF PORTB
     CALL DELAY
     MOVLW .7
     MOVWF COUNT1
LOP6     RLNCF PORTB   ; rotate portb 7 times to left
     CALL DELAY    ; with delay
     DECF COUNT1
     BNZ LOP6
     MOVLW B'10000000'
     MOVWF PORTB
     MOVLW .7
     MOVWF COUNT1
LOP7     RRNCF PORTB   ; rotate portb 7 times to right
     CALL DELAY    ; with delay
     DECF COUNT1
     BNZ LOP7
     CALL DELAY
     CLRF PORTB
     MOVLW B'10000000'
     MOVWF PORTC
     CALL DELAY
     MOVLW .7
     MOVWF COUNT1
LOP8     RRNCF PORTC   ; rotate portb 7 times to right
     CALL DELAY    ; with delay
     DECF COUNT1
     BNZ LOP8
     GOTO AGAIN    ; go to begining of this part
    ;THIRD PART
AGAIN2   BTFSS PORTA,1   ;checking portc RA1 pin
     BRA Start       ; if it is RESET go to START, otherwise continue
     MOVLW .8
     MOVWF COUNT1
LOP9     RLCF PORTC    ; rotate portB 7 times to LEFT
     BSF STATUS, C ; C=1
     CALL DELAY    ; with delay
     DECF COUNT1
     BNZ LOP9
     MOVLW .8
     MOVWF COUNT1
LOP10    RLCF PORTB    ; rotate portB 7 times to LEFT
     BSF STATUS, C ; C=1
     CALL DELAY    ; with delay
     DECF COUNT1
     BNZ LOP10
     MOVLW .8
     MOVWF COUNT1
LOP11    BCF STATUS, C ; C=0
     RRCF PORTB    ; rotate portB 7 times to RIGHT
     CALL DELAY    ; with delay
     DECF COUNT1
     BNZ LOP11
     MOVLW .8
     MOVWF COUNT1
LOP12    BCF STATUS, C ; C=0
     RRCF PORTC    ; rotate portB 7 times to right
     CALL DELAY    ; with delay
     DECF COUNT1
     BNZ LOP12
     GOTO AGAIN2  ; go to beginig of this part
    ;FORTH PART
AGAIN3   BTFSS PORTA,2   ; checking portc RA2 pin
     BRA Start       ; if it is RESET go to START, otherwise continue
         MOVLW .8
     MOVWF COUNT1
LOP13 BSF STATUS, C ; C=1
     RLCF PORTB    ; rotate portB 8 times to LEFT
     BSF STATUS, C ; C=1
     RRCF PORTC    ; rotate portC 8 times to RIGHT
     BSF STATUS, C ; C=1
     CALL DELAY
     DECF COUNT1
     BNZ LOP13
     MOVLW .7
     MOVWF COUNT1
LOP14    BCF STATUS, C   ; C=0
     RRCF PORTB      ; rotate portB 7 times to RIGHT
     BCF STATUS, C   ; C=0
     RLCF PORTC      ; rotate portC 7 times to LEFT
     BCF STATUS, C   ; C=0
     CALL DELAY
     DECF COUNT1
     BNZ LOP14
     GOTO AGAIN3    ; GO TO BEGINNINGOF THIS PART
     ;_____________Delay
DELAY    movlw .40  ; these loop values calculated to look better on proteus
     movwf 0x21
For5     movlw .50
     movwf 0x22
For4     movlw .1
     movwf 0x23
For3     nop
     nop
     DECF  0x23,f
     BNZ For3
     decf 0x22,f
     BNZ For4
     decf 0x21,f
     BNZ For5
     RETURN
     end
```

2.

```
     INPUT    equ 0x00 ; DEFINE LOCATIONS
H_BYTE   equ 0x10
L_BYTE   equ 0x11
C_Cont1  equ 0x30
C_Cont2  equ 0x31

     org 0h
     CLRF H_BYTE
     CLRF C_Cont1
     CLRF C_Cont2
     CLRF L_BYTE
     CLRF 0X01
     MOVLW .22 ; INPUT VALUE
     MOVWF INPUT
     MOVLW .1 ; move wreg 1
     MOVWF 0X01 ; First number is 1
     MOVWF L_BYTE ; Second number is 1
LOOP     BCF STATUS,0 ; clear carry
     ADDWF 0X01,W ; sum first and second number
     MOVWF 0X03   ; send summatıon to 0x03 location
     MOVFF L_BYTE,0X01 ; second number = first number
     MOVFF 0X03,L_BYTE ; summation = second number
     movf C_Cont1,W ; For carries
     addwfc C_Cont2,W
     movff C_Cont1,C_Cont2
     movwf C_Cont1
     movff C_Cont1,H_BYTE ; we moved carries to H_BYTE
     MOVF L_BYTE,W ; move second number to wreg
     DECF INPUT ; loop input times
     BNZ LOOP
     END
```

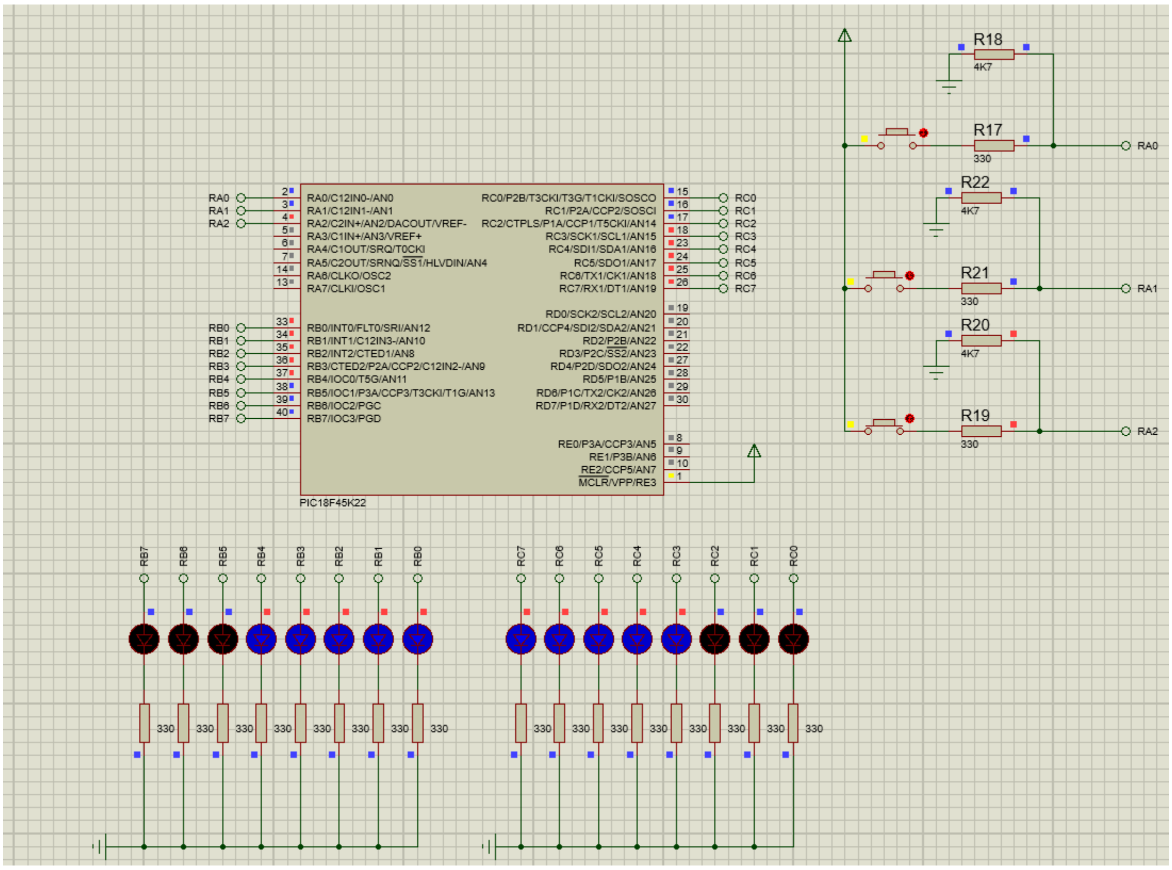
3.

```
     org 0h
     NUMBER1.1 EQU 0X30 ; first lower
     NUMBER1.2 EQU 0X62 ; first higher
     NUMBER2.1 EQU 0X2E ; second lower
     NUMBER2.2 EQU 0X43 ; second higher

     RESULT_4  EQU 0X10 ; MSB
     RESULT_3  EQU 0X11 ; 2.bit
     RESULT_2  EQU 0X12 ; 3.bit
     RESULT_1  EQU 0X13 ; LSB

     MOVLW NUMBER1.1 ; 1st low byte moved to wreg
     MULLW NUMBER2.1 ; 2nd x 1st low byte

     MOVFF PRODL,RESULT_1 ; result of low byte is directly to lowest byte
     MOVFF PRODH,0X20 ; result of higher byte is at 0x20 loc.

     MOVLW NUMBER1.2 ; 1st higher byte moved to wreg
     MULLW NUMBER2.1 ; 2nd low byte x 1st high byte

     MOVFF PRODL,0X22 ; result of low byte is at 0x22
     MOVFF PRODH,0X21 ; result of high byte is 0x21

     MOVLW NUMBER1.1 ; 1st lower byte moved to wreg
     MULLW NUMBER2.2 ; 1st low byte x 2nd high byte

     MOVFF PRODL,0X24  ; result of low byte is at 0x24
     MOVFF PRODH,0X23  ; result of high byte is 0x23

     MOVLW NUMBER1.2 ; 1st higher byte moved to wreg
     MULLW NUMBER2.2 ; 1st high byte x 2nd high byte

     MOVFF PRODL,0X25 ; result of low byte is at 0x25
     MOVFF PRODH,RESULT_4  ; result of high byte is directly to highest byte

     ; Summing numbers and carry controls
     MOVF 0X22,0
     ADDWF 0X24,F ; adding location 22h and location 24h
     BNC NO_CARRY0
     INCF 0x21,F ; ıf carry exist, +1
NO_CARRY0 MOVF 0X21,0
     ADDWF 0X23,F ; adding location 21h and location 23h
     BNC NO_CARRY1
     INCF RESULT_4,F ; ıf carry exist, +1

NO_CARRY1 MOVF 0X20,0
     ADDWFC 0X24,F ; adding location 20h and location 24h
     BNC NO_CARRY2
     INCF 0x25,F ; ıf carry exist, +1
NO_CARRY2 MOVF 0X25,0
     ADDWF 0X23,F ; adding location 25h and location 23h
     BNC NO_CARRY3
     INCF RESULT_4,F ; ıf carry exist, +1

NO_CARRY3 MOVFF 0X23,RESULT_3 ; moving result location
      MOVFF 0X24,RESULT_2 ; moving result location
    END
```

EXPLANATIONS:

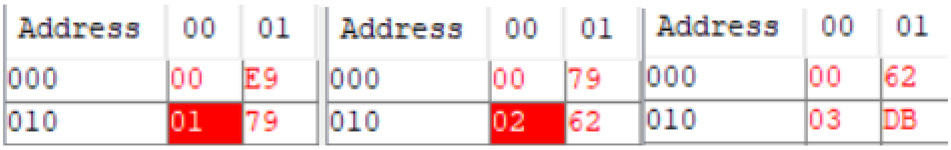
QUESTION-1 :

Our aim is to simulate Knight Rider LED scanner on each LED from pin 0 to 7 of PORTB and PORTC and to do these visual effects, we should use the commands for rotating the bits such as RRNCF, RRCF or RLCF.



QUESTION-2 :

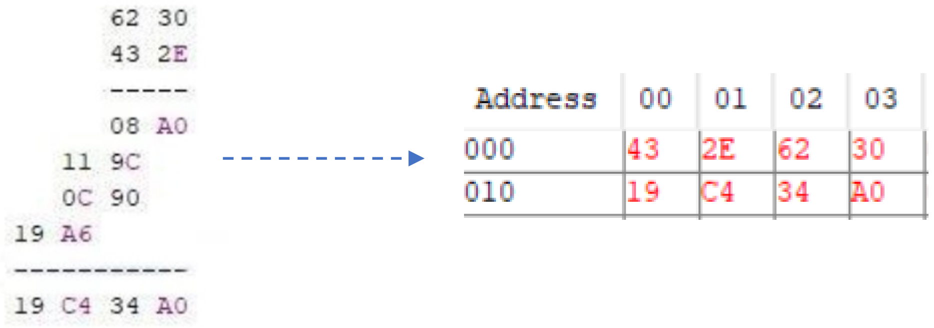
Our aim is to implement the Fibonacci sequence in this question. You can see my 14th, 15th and 16th result of Fibonacci sequence below. Location 0x10 is higher byte and location 0x11 is lower byte.



Address	00	01	Address	00	01	Address	00	01
000	00	E9	000	00	79	000	00	62
010	01	79	010	02	62	010	03	DB

QUESTION-3 :

Our aim is to write code for multiplying two 16 bit signed number. You can see my result on the location 0x10, 0x11, 0x12, and 0x13.



```
      62 30
      43 2E
      -----
      08 A0
   11 9C
   0C 90
19 A6
-----------
19 C4 34 A0
```

Address	00	01	02	03
000	43	2E	62	30
010	19	C4	34	A0

Note:

This document will be prepared before the lab session. Unless you bring this document in the desired format or prepared, you will not be let to the session.