

Exo If the buttons connected RA1 & RA2 are pressed the LED connected to RB0 is turned ON.

Sln:

```

list p=16f84A
include "16f84A.INC"
clr PORTB;
bsf STATUS, RPO;
movlw 0xFF;
movwf TRISA;
clr TRISB;
bcf STATUS, RPO;

```

```

test-PORTA    movlw b'00011001';
               xorwf PORTA, W;
               btfss STATUS, Z;
               goto test-PORTA
loop          goto loop

```

### Arithmetic Operation Commands

addlw, addwf, sublw, subwf, rlf rrf

addlw k; → Add Literal into W

$$(W) \leftarrow (W) + k$$

The contents of the W register are added to the eight-bit literal 'k' and the result is placed in the W register.

2

addw f, d; → Add word f

If  $d=0 \rightarrow (w) \leftarrow (w) + (f)$

$d=1 \rightarrow (f) \leftarrow (w) + (f)$

Status affected: C, DC, Z

Add the contents of the W register with register 'f'. If  $d=0$ , the result is stored in the W register. If  $d=1$ , the result is stored back in register 'f'.

Summation of two 8-bit Numbers.

Exo

Num1 = 0x5A

Num = Num1 + Num2 ?

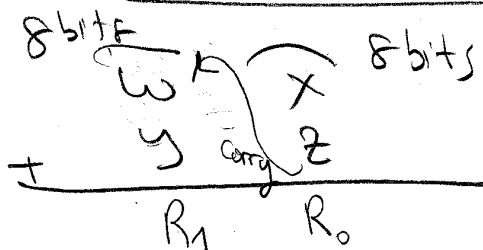
Num2 = 0x53

Sln: movlw 0x5A

addlw 0x53

movwf PORTB → result is displayed in PORTB

Summation of two 16-bit Numbers.

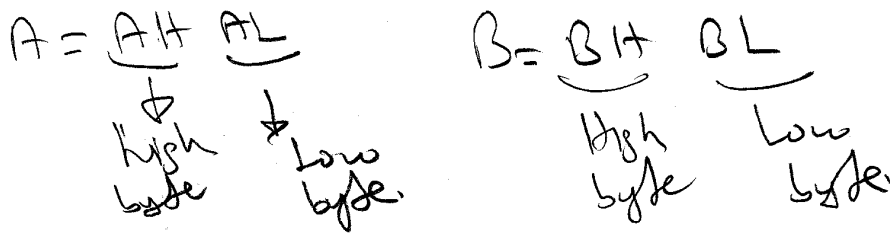


Q.3

Num1 + Num2 = ?  
16-bit    16-bit

first lower bytes are summed  
then upper " " " including  
the carry coming from lower byte  
- summation

Let Numbers be A & B



Let A = 0x0213 → AH=02 AL=13  
     B = 0x2EE3 → BH=2E BL=E3

```
list P=16f84A
include "16f84A.INC"
c1rf PORTB;
bsf STATUS, RPO;
c1rf TRISB;
movlw 0xFF;
movwf TRISA;
bcf STATUS, RPO;

AL equ 0x0C
AH equ 0x0D
BL equ 0x0E
BH equ 0x0F
```

4

```
movlw 0x13;  
movwf AL;  
movlw 0x02;  
movwf AH;  
movlw 0xE3;  
movwf BL;  
movlw 0x2E;  
movwf BH;
```

Sum

```
movf AL,W,  
addwf BL,F;  
btfsc STATUS,C;  
incf BH,F;  
movf AH,W,  
addwf BH,W;
```

display-low-byte

```
movf BL,W;  
movwf PORTB;
```

wait-for-upper-disp.

```
btsfc PORTA,1
```

```
goto wait-for-upper-disp
```

display-high-byte

```
movf BH,W;  
movwf PORTB;
```

loop

```
goto loop
```

```
end
```