

### **To be done in Edsim51 Simulator:**

1. Suppose you are given 16 bit binary data, you are required to compute the number of 1's that the binary data have and display on a seven segment display.

ORG 0H

; Initialize variables

MOV R0, #0H ; Counter for the number of 1's

MOV DPTR, #DATA ; Load address of binary data into DPTR

MOV R1, #16 ; Counter for number of bits

LOOP:

MOVX A, @DPTR ; Load next byte of binary data into accumulator

INC DPTR ; Move to the next byte

CLR C ; Clear carry flag

JNB ACC.0, SKIP ; Jump if the least significant bit is 0

INC R0 ; Increment counter if the least significant bit is 1

SKIP:

DJNZ R1, LOOP ; Decrement R1 and jump to LOOP if it's not zero

; Convert the count to BCD (Binary Coded Decimal)

MOV A, R0 ; Move the count to accumulator

MOV B, #10 ; Divisor for BCD conversion

DIV AB ; Divide count by 10

MOV R2, A ; Store the tens place in R2

MOV A, B ; Move the quotient to A

MOV R3, A ; Store the ones place in R3

; Display the count on a seven-segment display

```

MOV P1, R2      ; Display tens place on P1
CALL DELAY      ; Delay for a short period
MOV P1, R3      ; Display ones place on P1
CALL DELAY      ; Delay for a short period
SJMP $          ; End of program, loop indefinitely

```

DELAY:

; Insert delay subroutine here (implement based on your specific requirements)

RET

DATA:

; Your 16-bit binary data goes here

END

1. Write a program to generate a square wave of 2 KHz frequency on Pin 1.5. You are required to use a timer in mode 1. Assume that XTAL=11.0592MHz.

ORG 0H

; Initialize Timer 1 in Mode 1

```
MOV TMOD, #10H ; Set Timer 1 to Mode 1 (16-bit auto-reload)
```

```
MOV TH1, #0FDH ; Load Timer 1 high byte with initial value for 2 KHz
```

```
MOV TL1, #0FDH ; Load Timer 1 low byte with initial value for 2 KHz
```

; Set up Pin 1.5 for output

```
SETB P1.5      ; Set Pin 1.5 as output
```

; Start Timer 1

```
SETB TR1        ; Start Timer 1
```

MAIN\_LOOP:

```
; Your main program loop here  
SJMP MAIN_LOOP ; Infinite loop
```

```
END
```

2. Write a program that continuously gets 8-bit data from P0 and sends it to P1 while simultaneously creating a square wave of 200 micro sec period on pin P2.1. Use timer1 to create the square wave. Assume that XTAL=11.0592MHz.

```
ORG 0H
```

```
; Define constants
```

```
XTAL_FREQ equ 11059200 ; XTAL frequency
```

```
SQUARE_WAVE_PERIOD equ 200 ; Square wave period in microseconds
```

```
; Calculate Timer 1 reload value for desired square wave period
```

```
TIMER1_RELOAD_VAL equ (65536 - (XTAL_FREQ / 12 / 1000000 *  
SQUARE_WAVE_PERIOD))
```

```
; Initialize Timer 1 in Mode 1
```

```
MOV TMOD, #10H ; Set Timer 1 to Mode 1 (16-bit auto-reload)
```

```
; Load Timer 1 reload value
```

```
MOV TH1, HIGH(TIMER1_RELOAD_VAL) ; Load Timer 1 high byte with  
reload value
```

```
MOV TL1, LOW(TIMER1_RELOAD_VAL) ; Load Timer 1 low byte with  
reload value
```

; Set up Pin P2.1 for output

SETB P2.1 ; Set Pin P2.1 as output

; Start Timer 1

SETB TR1 ; Start Timer 1

MAIN\_LOOP:

; Read 8-bit data from Port 0 (P0) and send it to Port 1 (P1)

MOV A, P0 ; Read data from P0

MOV P1, A ; Send data to P1

; Wait for Timer 1 overflow to create square wave

WAIT\_FOR\_TIMER1\_OVERFLOW:

JNB TF1, WAIT\_FOR\_TIMER1\_OVERFLOW ; Wait until Timer 1 overflows (square wave period elapsed)

CLR TF1 ; Clear Timer 1 overflow flag

; Toggle Pin P2.1 to generate square wave

CPL P2.1 ; Toggle Pin P2.1

; Continue looping

SJMP MAIN\_LOOP

END