Interfacing LED/LEDS:

1. Flash an LED connected to any port line at the rate of 1 second for 10 times and stop.

2. Write an ECP to display below mentioned patterns on 8-led’s (4-active high led’s and 4-active low led’s)

   a. Switch ON one by one led from left to right at the rate of 1 second

   b. Switch ON one by one led from right to left at the rate of 1 second

3. Create a display on 8 LED’s (4-active high and 4-active low) connected to port0 (0.0-0.7) in the given pattern.
   a. First all LEDs should be OFF, at 1st second, LED 0 and LED 7 (P0.0 and P0.7) should be ON
   b. At 2nd second, LED 1 and LED 6 (P0.1 and P0.6) should be ON
   c. At 3rd second, LED 2 and LED 5 (P0.2 and P0.5) should be ON
   d. At 4th second, LED 3 and LED 4 (P0.3 and P0.4) should be ON
   e. Repeat above pattern for 5 times and stop the process.

4. Write an ECP to display binary equivalent of 0-255 on 8-leds (4-active high led’s and 4-active low led’s)

5. Write an ECP to find the second highest digit in a given integer and display its binary equivalent on 8-LEDS (4-Active High LEDS& 4-Active Low LEDS).

6. Write an ECP to display binary equivalent of prime numbers within the range of 0-50 @ 5 seconds. (Note: Use IsPrime() function)

7. Write an ECP to print the binary equivalent of odd digits in a given integer on 8 LEDS @ 5 seconds.

8. Write an ECP to print the binary equivalent of sum of the digits for the given integer on LEDS (4-Active High LEDS& 4-Active Low LEDS).

9. Write an ECP to print the binary equivalent of product of the digits for the given integer on LEDS (4-Active High LEDS& 4-Active Low LEDS).

   Note: if the product is >255, Switch on alternate LEDS.

10. Write an ECP to print the binary equivalent of power of 2 numbers with in the range of 0-255 @ 5 seconds.

Interfacing LED & SWITCH:

1. Create a display on 8 LED's (4-active high and 4-active low) using 4 active low switches connected to port0 (0.16-0.19) in the pattern explained:

   a. First all LEDs should be OFF.

   b) if 1st switch is pressed, LED 0 and LED 7 (P0.0 and P0.7) should be ON

   c) if 2nd switch is pressed, LED 1 and LED 6 (P0.1 and P0.6) should be ON

   d) if 3rd switch is pressed, LED 2 and LED 5 (P0.2 and P0.5) should be ON

   e) if 4th switch is pressed, LED 3 and LED 4 (P0.3 and P0.4) should be ON

2. Write an ECP to display binary equivalent of switch press count on 8-leds (4-active high leds and 4-active low leds)

3. PART1: Implement up and down counter. Use two switches and 8 leds (4-active high leds and 4-active low leds) as mentioned**:** sw1 for incrementing count, sw2 for decrement count and display updated count on leds.

Note: if count value is 0, at this time if sw2 pressed it should display 0 only on leds and if count value is 255 then if sw1 pressed it should display 255 only on leds.

4. Write an ECP to print the binary equivalent of switch press count within 500 milli seconds.

Note: Time slice should start after the first switch press

5. PART2: Implement up and down counter. Use two AL switches and 8 leds (4-active high leds and 4-active low leds) as mentioned**:**

As long as sw1 is pressed increment the count value with respect to 1 sec

As long as sw2 is pressed decrement count value with respect to 1 sec

And display updated count on leds.

If both switches are pressed at a time, don't do any operation on count.

Note: if count value is 0, at this time if sw2 pressed it should display 0 only on leds and if count value is 255 then if sw1 pressed it should display 255 only on leds.

6. Write an ECP to fulfill the below mentioned operations.

   1. Initially all leds should be off.
   2. Display even numbers on 8 leds with respect to 1 second if sw1 is pressed. Continue this process until sw2 is pressed.
   3. Display odd numbers on 8 leds with respect to 1 second if sw2 is pressed. Continue this process until sw1 is pressed.

      Note: if number is >255 then roll over to 0 and repeat the operation.

Seven Segment Display + Switch:

1. Write an ECP to test seven segment display (all segments) is working or not.

2. Write an ECP to Show up counting from 0 to 9 on seg1 and then after show down counting 9 to 0 on seg2 @ of 1sec (approximately).

3. Write an ECP to display numbers 0-9 on two segments @ 1sec in the below given format.

Even numbers on segment 1 and odd numbers on segment2

4. Write an ECP to Show up counting from 00 to 99 on two multiplexed seven segments @ of 1sec (approximately).

5. Write an ECP to Show down counting from 99 to 00 on two multiplexed seven segments @ of 1sec (approximately).

6. Display 00 on two multiplexed segments and make it flash for 5 times and then display 88 and make it flash for 5 times then stop.

7. Write an ECP to display the switch press count on two multiplexed seven segment display. (Note: if switch is pressed, need to display the updated count value after switch is released)

8. Implement a 2 digit up and down counter. Use two switches as mentioned: sw1 - incrementing count & sw2 - decrement count.

Note: if count value is 0, then if sw2 pressed it should display 00 only and if count value is 99 then if sw1 pressed it should display 99 only.

9. Display float values from 0.0 to 9.9 on two multiplexed seven segment display.

10. Write an ECP for Single segment only should be lit (starting from segment A) cyclically around the periphery of single seven segment (5 times clock wise and 5 times anti-clock wise @ proper visualization).

11. Write an ECP to print the switch press count within 500 milli seconds on two multiplexed seven segment displays.

Note: Time slice should start after the first switch press

Explanation: Initially 0 will display on two multiplexed displays. Once switch is pressed, then time slice will start. If switch is pressed with in time slice again then time slice starts from 0. If there is no switch press within time slice then display the updated count on two multiplexed seven segment displays. After switch is pressed, within time slice no need to display any value on segments. (This logic is giving support for SMS keypad program implementation)

12. Produce the display on seven segments using switches connected to any port in the pattern explained:

    a. First two Segments should be OFF
    b. For every SW1 press the value should be incremented and displayed on the seven segments.
    c. For every SW2 press the value should be decremented and displayed on the seven segments.
    d. For every SW3 press the value should be reset.

13. Write an ECP to display the prime numbers within 0-99 on two multiplexed seven segment displays. (Note: must use IsPrime() function)

14. Write an ECP to display the Fibonacci series numbers (0-99) on two multiplexed seven segment displays.

15. Write an ECP to display the 10 random numbers with in 0-99 on two multiplexed seven segment display with respect to 5 seconds.

16. Write an ECP to implement electronic dice using one switch and single segment display.

17. Write an ECP to display 10 random numbers within 0-9 and display the sum of all random numbers on two multiplexed seven segment displays.

    Note: display random numbers also.

Interfacing 16X2 LCD:

1. Write an ECP to develop the driver for 16*2 alphanumeric LCD.

    a) To display a character

    b) To display a string

    c) To display an integer

    d) To display float number up to three decimal places.

    e) To display any custom character.

2. Write a program to display the message “VECTOR” on the first Line and “Institute” on the second line of a 2x16 LCD. Then make “Institute” flash at the rate of 1sec for 5 times, then clear the LCD screen

3. Write an ECP to rotate a string on LCD (From right to left)

4. Write an ECP display the basic time (HH:MM:SS) on LCD.

    Note: don’t use RTC registers

5. Write an ECP to display the switch press count on LCD.

6. Write an ECP to take 20 numbers randomly in an array and find the prime numbers in the list of numbers to display on LCD.

    Note: must use rand() function

7. Write an ECP to take ten 3-digit numbers randomly in an array and find palindrome numbers in the list of numbers to display on LCD.

Note: must use rand() function

8. Write an ECP to check whether given string is palindrome or not and display result on LCD.

9. Write a program to display the message “VECTOR” on the first Line and “Institute” on the second line of a 2x16 LCD. Then make “Institute” scroll from right to left on second line of LCD screen.

10. Write an ECP to convert first character of each word in to upper case and display the final result on LCD.

Note: Connect two switches. Based on first switch press, take “vector india” as input and do the operation; Based on second switch press, take “vector Hyderabad” as input and do the operation. In both cases print input and output on lcd screen.

Output:

When Switch1 is pressed, display “vector india” on the 1st line of LCD and when the switch1 is released display “Vector India” on the 2nd line of LCD.

When Switch2 is pressed, display “vector hyderabad” on the 1st line of LCD and when the switch2 is released display “Vector Hyderabad” on the 2nd line of LCD.

If none of the switch is pressed, display “waiting for input” on the 1st line of LCD.

11. Write an ECP to display the ASCII table information for A-Z, a-z & 0-9 on LCD screen with respect to 5 seconds.

For example,

```
1st line:  A    D    H    O
2nd line:  A    65   41   101
```

After 5 secs

```
1st line:  A    D    H    O
2nd line:  B    66   42   102 and so on ….
```

12. Write an ECP to display the binary value for A-Z, a-z & 0-9 on LCD screen with respect to 5 seconds.

For example,

1st line: A

2nd line: 01000001

After 5 secs

1st line: B

2nd line: 01000010 and so on ….

13. Write an ECP to convert first and last character of each word in to upper case and display the final result on LCD.

Note: Connect two switches. Based on first switch press, take “vector india” as input and do the operation; Based on second switch press, take “vector Hyderabad” as input and do the operation. In both cases print input and output on lcd screen.

Output:

When Switch1 is pressed, display “vector india” on the 1st line of LCD and when the switch1 is released display “VectoR IndiA” on the 2nd line of LCD.

When Switch2 is pressed, display “vector hyderabad” on the 1st line of LCD and when the switch2 is released display “VectoR HyderabaD” on the 2nd line of LCD.

If none of the switch is pressed, display “waiting for input” on the 1st line of LCD.

14. Write an ECP to remove extra spaces in a given string and display the final result on LCD.

Note: Connect two switches. Based on first switch press, take “ An Apple a day” as input and do the operation; Based on second switch press, take “ vector   stu !” as input and do the operation. In both cases print input and output on lcd screen.

Output:

When Switch1 is pressed, display “ An Apple a day” on the 1st line of LCD and when the switch1 is released display “An Apple a day” on the 2nd line of LCD.

When Switch2 is pressed, display “ vector   stu !” on the 1st line of LCD and when the switch2 is released display “vector stu !” on the 2nd line of LCD.

If none of the switch is pressed, display “waiting for input” on the 1st line of LCD.

15. Write an ECP to develop any simple game on LCD by using customised characters.

Note: if required you can take any input devices (switch, keypad) support

Assignments on 4x4 matrix keypad with output devices:

1. Write an ECP for displaying binary equivalent of a key Value from 4X4 matrix keypad on 8-LEDS (4-active high leds & 4-active low leds).

2. Write an ECP for displaying a key Value from 4X4 matrix keypad on 2-multiplexed seven segment displays.

3. Write an ECP for implementing a basic calculator using 4X4 matrix keypad and 2-multiplexed seven segment displays.

4. Write an ECP for implementing a basic calculator using 4X4 matrix keypad and 16x2 lcd.

Note: 1+2=3, 3*8=24, 3-6=-3 etc …

5. Write an ECP for implementing multi digit calculator using 4X4 matrix keypad and 16x2 lcd.

Note: 123+23=146, 33*38=1254, 345-627=-282 etc …

6. Write an ECP to access a device (door/fan/bulb) with a valid password entry from 4x4 matrix keypad.

Note: must include all test cases which are giving support for real-time applications

7. Write an ECP to design a menu-based calculator using 4x4 matrix keypad and 16X2 lcd.

Menu: 1. Addition 2. Subtraction …. etc

8. Write an ECP to check whether given number is prime or not and display result on LCD.

Note: use keypad to take the input (multi digit) and must use IsPrime() & ReadNum() functions

9. Write an ECP to take one number (multi digit) from keypad and display its binary equivalent on LCD.

Note: must use BinLCD() function

10. Write an ECP to check whether given number is palindrome or not and display result on LCD.

Note: use keypad to take the input and must use IsPalindrome() function

Assignments on UART:

1. Write an ECP to develop the driver for on-chip UART.

   a) To display a character

   b) To display a string

   c) To display an integer

   d) To display float number up to three decimal places.

2. Write an ECP for transmitting string through UART and display it on hyper terminal with 9600 baud rate.

   Example: "Welcome to VECTOR"

3. Write an ECP for receiving one character from PC through UART with 9600 baud rate and display its binary equivalent on 8 – LEDS (4 – Active high leds & 4 – Active low leds).

4. Write an ECP to infinitely receive any character then re-transmit its ASCII value in decimal and hex format. Output should be seen as for example below

Output: if Received character is 'A', then output should be 65 0x41

If Received character is 'a', then output should be 97 0x61

If Received character is 'Z', then output should be 90 0x5A and so on …

5. Write an ECP to infinitely receive any character from PC through UART and display that character on LCD 1st line 1st position.

6. Write an ECP to infinitely receive any string from PC through UART and display that string on LCD 1st line 1st position.

7. Write an ECP for transmitting characters A-Z from LPC21xx based CPU to the hyper terminal of PC through UART for every switch press from user.

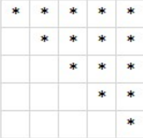
8. Write an ECP to accept valid password from user to access security-based devices using UART with 9600 baud rate.

Note: password should contain,

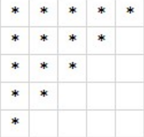
At least one numeric digit, at least one Lowercase Letter, at least one Uppercase Letter, at least one special letter, must not have space (' '), must not have slash ('/') and at least 8 characters.

If someone inputs an invalid password, the code prints: "Invalid Password" on hyper terminal otherwise, it prints: "Valid Password".

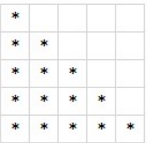
9. Write an ECP to display the below mentioned patterns on hyper terminal with the help of UART0 with 19200 baud rates.



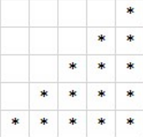
*	*	*	*	*
	*	*	*	*
		*	*	*
			*	*
				*



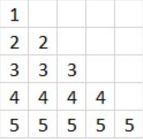
*	*	*	*	*
*	*	*	*	
*	*	*		
*	*			
*				



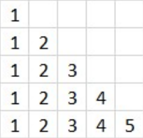
*				
*	*			
*	*	*		
*	*	*	*	
*	*	*	*	*



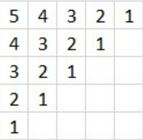
				*
			*	*
		*	*	*
	*	*	*	*
*	*	*	*	*



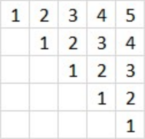
1				
2	2			
3	3	3		
4	4	4	4	
5	5	5	5	5



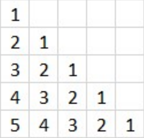
1				
1	2			
1	2	3		
1	2	3	4	
1	2	3	4	5



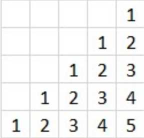
5	4	3	2	1
4	3	2	1	
3	2	1		
2	1			
1				



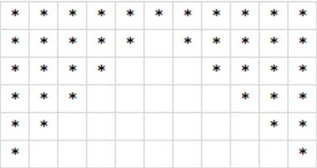
1	2	3	4	5
	1	2	3	4
		1	2	3
			1	2
				1



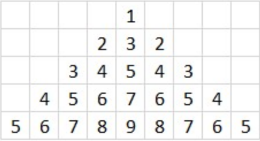
1				
2	1			
3	2	1		
4	3	2	1	
5	4	3	2	1



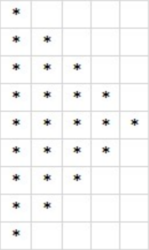
				1
			1	2
		1	2	3
	1	2	3	4
1	2	3	4	5



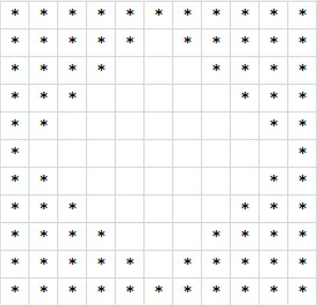
*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*		*	*	*	*	*
*	*	*	*				*	*	*	*
*	*	*						*	*	*
*	*								*	*
*										*



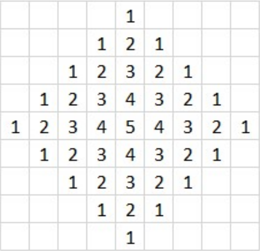
				1				
			2	3	2			
		3	4	5	4	3		
	4	5	6	7	6	5	4	
5	6	7	8	9	8	7	6	5



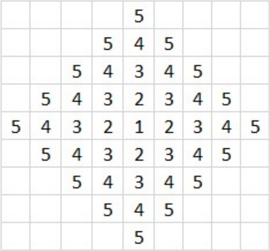
*				
*	*			
*	*	*		
*	*	*	*	
*	*	*	*	*
*	*	*	*	
*	*	*		
*	*			
*				



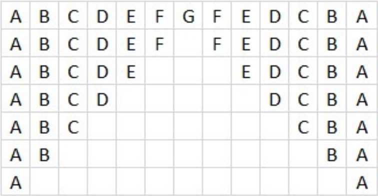
*	*	*	*	*	*	*	*	*	*	*
*	*	*	*	*		*	*	*	*	*
*	*	*	*				*	*	*	*
*	*	*						*	*	*
*	*								*	*
*										*
*	*								*	*
*	*	*						*	*	*
*	*	*	*				*	*	*	*
*	*	*	*	*		*	*	*	*	*
*	*	*	*	*	*	*	*	*	*	*



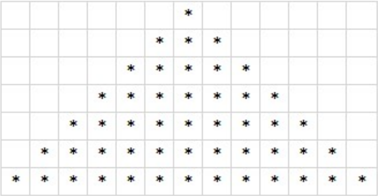
				1				
			1	2	1			
		1	2	3	2	1		
	1	2	3	4	3	2	1	
1	2	3	4	5	4	3	2	1
	1	2	3	4	3	2	1	
		1	2	3	2	1		
			1	2	1			
				1				



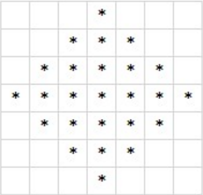
				5				
			5	4	5			
		5	4	3	4	5		
	5	4	3	2	3	4	5	
5	4	3	2	1	2	3	4	5
	5	4	3	2	3	4	5	
		5	4	3	4	5		
			5	4	5			
				5				



A	B	C	D	E	F	G	F	E	D	C	B	A
A	B	C	D	E	F		F	E	D	C	B	A
A	B	C	D	E				E	D	C	B	A
A	B	C	D						D	C	B	A
A	B	C								C	B	A
A	B										B	A
A												A



						*						
					*	*	*					
				*	*	*	*	*				
			*	*	*	*	*	*	*			
		*	*	*	*	*	*	*	*	*		
	*	*	*	*	*	*	*	*	*	*	*	
*	*	*	*	*	*	*	*	*	*	*	*	*



			*			
		*	*	*		
	*	*	*	*	*	
*	*	*	*	*	*	*
	*	*	*	*	*	
		*	*	*		
			*			