CODE:

1]……// YES_LCD_SKETCH_10_14_12

```
#include <LiquidCrystal.h>
//lcd(RS, E, D4, D5, D6, D7)
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
int numRows = 2;
int numCols = 16;

void setup()
{
  Serial.begin(9600);
  lcd.begin(numRows, numCols);
  lcd.clear();                          // clears the display
  lcd.setCursor(0,0);                   // sets the position of cursor in row and column
  lcd.print("Thank God!!!");
  lcd.setCursor(0,1);
  lcd.print("This WORKS!");
}

  void loop()
  {
    if (Serial.available() > 0)
    {
      char ch = Serial.read();
      if (ch =='#')
      {
        lcd.clear();
      }
      else if (ch == '/')
      {
        // new line
        lcd.setCursor (0,1);
      }
      else
      {
        lcd.write(ch);
      }
    }
  }
```

2]…… //YES_12-5_1 8 PIN MODE 2 PORTS 10-15-12

```
#define F_CPU 16000000UL
#include <LiquidCrystal.h>
#include <avr/io.h>
#include <util/delay.h>
//lcd(RS, E, D0, D1,D2,D3,D4,D5,D6,D7)
LiquidCrystal lcd(8,9,0,1,2,3,4,5,6,7);
#define LCD_DPRT  PORTD                 // does NOT work with port A
#define LCD_DDDR  DDRD                  // ONLY with all B or C or a
#define LCD_DPIN  PIND                  // combination of the two!!!
#define LCD_CPRT  PORTB
#define LCD_CDDR  DDRB
#define LCD_CPIN  PINB
```

```
#define LCD_RS  8
#define LCD_RW  1
#define LCD_EN  9

void setup()
{
  Serial.begin(9600);
}
//*******************************************************
void delay_us(unsigned int d)
{
  _delay_us(d);
}
//*******************************************************
void lcdCommand( unsigned char cmnd )
{
LCD_DPRT = cmnd;                    // ready data lines
LCD_CPRT &= ~ (1<<LCD_RS);          // RS low (lower line on the timing diagram) to select the Instruction register.
LCD_CPRT &= ~ (1<<LCD_RW);          // RW low to write instructions.
LCD_CPRT |= (1<<LCD_EN);            // Enable pin high to latch data on the falling edge.
delay_us(1);                        // Enable pin high for 1us to enable latching of instruction set.
LCD_CPRT &= ~ (1<<LCD_EN);          // LCD_EN pin of Port B low after sending an instruction.
delay_us(100);                      // Enable pin low for 100 us after sending instruction set.
}
//*******************************************************
void lcdData( unsigned char data )
{
LCD_DPRT = data;
LCD_CPRT |= (1<<LCD_RS);            // RS pin high (upper line on the timing diagram)to select the Data register.
LCD_CPRT &= ~ (1<<LCD_RW);          // RW low to write instructions.
LCD_CPRT |= (1<<LCD_EN);            // Enable pin set to latch data on the falling edge.
delay_us(1);                        // Enable pin high for 1us to enable latching of data set.
LCD_CPRT &= ~ (1<<LCD_EN);          // LCD_EN pin of Port B low after sending data byte.
delay_us(100);                      // Delay 100 us between data sent.

//*******************************************************
void lcd_init()
{
  LCD_DDDR = 0xFF;
  LCD_CDDR = 0xFF;
  LCD_CPRT &=~(1<<LCD_EN);
  delay_us(2000);
  lcdCommand(0x38);
  lcdCommand(0x0E);
  lcdCommand(0x01);
  delay_us(2000);
  lcdCommand(0x06);
}
//*******************************************************
void lcd_gotoxy(unsigned char x, unsigned char y)
{
 unsigned char firstCharAdr[]={0x80,0xC0,0x94,0xD4}; //table 12-5
 lcdCommand(firstCharAdr[y-1] + x - 1);
```

```
 delay_us(100);
}
//*****************************************************
void lcd_print( char * str )
{
  unsigned char i = 0 ;
  while(str[i]!=0)
  {
    lcdData(str[i]);
    i++ ;
  }
}
//*****************************************************
int main(void)
  {
          lcd_init();
          lcd_gotoxy(1,1);
          lcd_print("Thank God!");
          lcd_gotoxy(1,2);
          lcd_print("This compiles!");
          while(1);
          return 0;
  }
```

3] (Same as 2] above except for the lcdCommand and lcdData functions

……

```
void lcdCommand( unsigned char cmnd )
{
LCD_DPRT = cmnd;                    // ready data lines
LCD_CPRT &= ~ (1<<LCD_RS);          // RS low (lower line on the timing diagram)to select the Instruction register.
LCD_CPRT &= ~ (1<<LCD_RW);          // RW low to write instructions.
delay_us(.020);                     // Delay 20 ns for address set-up time.
LCD_CPRT |= (1<<LCD_EN);            // Enable pin high to latch data on the falling edge.
delay_us(.23);                      // Enable pin high for 230 ns to enable latching of instruction set.

LCD_CPRT &= ~ (1<<LCD_EN);          // LCD_EN pin of PortB low after sending an instruction.
delay_us(.23);                      // Enable pin low for 230 ns after sending instruction set.
}
void lcdData( unsigned char data )
{
LCD_DPRT = data;
LCD_CPRT |= (1<<LCD_RS);            // RS pin high (upper line on the timing diagram)to select the Data register.
LCD_CPRT &= ~ (1<<LCD_RW);           // RW low to write instructions.
delay_us(.020);                     // Delay 20 ns for address set-up time.
LCD_CPRT |= (1<<LCD_EN);            // Enable pin set to latch data on the falling edge.
delay_us(.23);                      // Enable pin high for 230 ns to enable latching of data set.

LCD_CPRT &= ~ (1<<LCD_EN);          // LCD_EN pin of PortB low after sending data byte.
delay_us(.23);                      // Delay 230 ns between data sent.
}
……..
```

4] 12.7.1 4 bit mode, 1port

```
//<CODE6—12.7.1:    4bit mode, 1port>
(SAME AS BEFORE)
          }
void delay_us(float d)
{
  _delay_us(d);
}
void delay_ms(float d)
{
  _delay_ms(d);
}
//*****************************
void lcdCommand( unsigned char cmnd ){
  LCD_PRT = (LCD_PRT & 0x0F) | (cmnd & 0xF0);
  LCD_PRT &= ~ (1<<LCD_RS);                    //RS = 0 for command
  LCD_PRT &= ~ (1<<LCD_RW);                    //RW=0 for write
  LCD_PRT |= (1<<LCD_EN);                      //EN =1 for H-to-L
  delay_us(1);                                 //wait to make EN wider
  LCD_PRT &= ~ (1<<LCD_EN);

  delay_us(20);                                //wait

  LCD_PRT = (LCD_PRT & 0x0F) | (cmnd << 4);    //EN=1for H-to-L
  LCD_PRT |= (1<<LCD_EN);                      //wait to make EN wider
  delay_us(1);                                 //EN=0 for H-to-L
  LCD_PRT &= ~ (1<<LCD_EN);
}
//*****************************
void lcdData( unsigned char data )
{
  LCD_PRT = (LCD_PRT & 0x0F) | (data & 0xF0);
  LCD_PRT |= (1<<LCD_RS);                      // RS = 1 for data
  LCD_PRT &= ~ (1<<LCD_RW);                    //RW= 0 for write
  LCD_PRT |= (1<<LCD_EN);                      //EN=1 for H-to-L
  delay_us(1);                                 //wait to make EN wider
  LCD_PRT &= ~ (1<<LCD_EN);                    //EN=0 for H-to-L

  LCD_PRT = (LCD_PRT & 0x0F) | (data << 4);
  LCD_PRT |= (1<<LCD_EN);                      //EN=1 for H-to-L
  delay_us(1);                                 //wait to make EN wider
  LCD_PRT &= ~ (1<<LCD_EN);                    //EN=0 for H-to-L

}
//*****************************
void lcd_init(){
  LCD_DDR = 0xFF;                              // LCD port is output
  LCD_PRT &=~(1<<LCD_EN);                      //LCD_EN=0
  delay_us(2000);                              //wait for stable power
  lcdCommand(0x33);                            // $33 for 4 bit mode
  delay_us(100);                               //wait
  lcdCommand(0x32);                            //$32 for 4 bit mode
```

```
  delay_us(100);                                  //wait
  lcdCommand(0x28);                               //$28 for 4 bit mode
  delay_us(100);                                  //wait
  lcdCommand(0x0e);                               //$32 for 4 bit mode
  delay_us(100);                                  //wait
  lcdCommand(0x01);                               //display on, cursor on wait
  delay_us(2000);                                 //wait
  lcdCommand(0x06);                               //shift cursor right
  delay_us(100);
}
//****************************
void lcd_gotoxy(unsigned char x, unsigned char y)   //table 12-5
{    unsigned char firstCharAdr[] = {0x80, 0xC0, 0x94, 0xD4};
  lcdCommand(firstCharAdr[y-1] + x - 1);
  delay_us(100);
}
//************************
void lcd_print( char * str )
{
  unsigned char i = 0 ;

  while(str[i]!=0)
  {
    lcdData(str[i]);
    i++ ;
  }
}
//*****************************
int main(void)
{
  lcd_init();
  while(1)                                        // stay forever
  {
          lcd_gotoxy(1,1);
          lcd_print("The world is but");
          lcd_gotoxy(1,2);
          lcd_print("one country");
          delay_ms(1000);

          lcd_gotoxy(1,1);
          lcd_print("and this one");
          lcd_gotoxy(1,2);
          lcd_print("also compiles");
          delay_ms(1000);
  }
  return 0;
}
//*********************************************
//*********************************************
```

/CODE