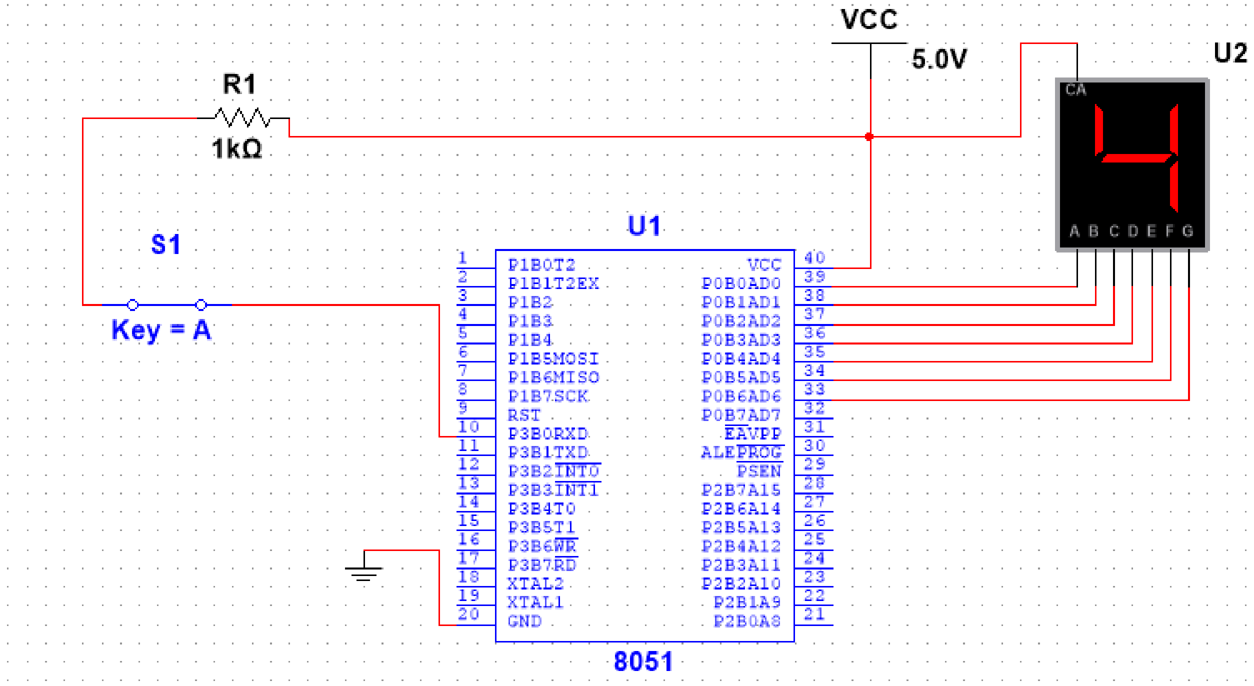
ASSIGNMENT 2

QUESTION: 1



ASSEMBLY CODE:

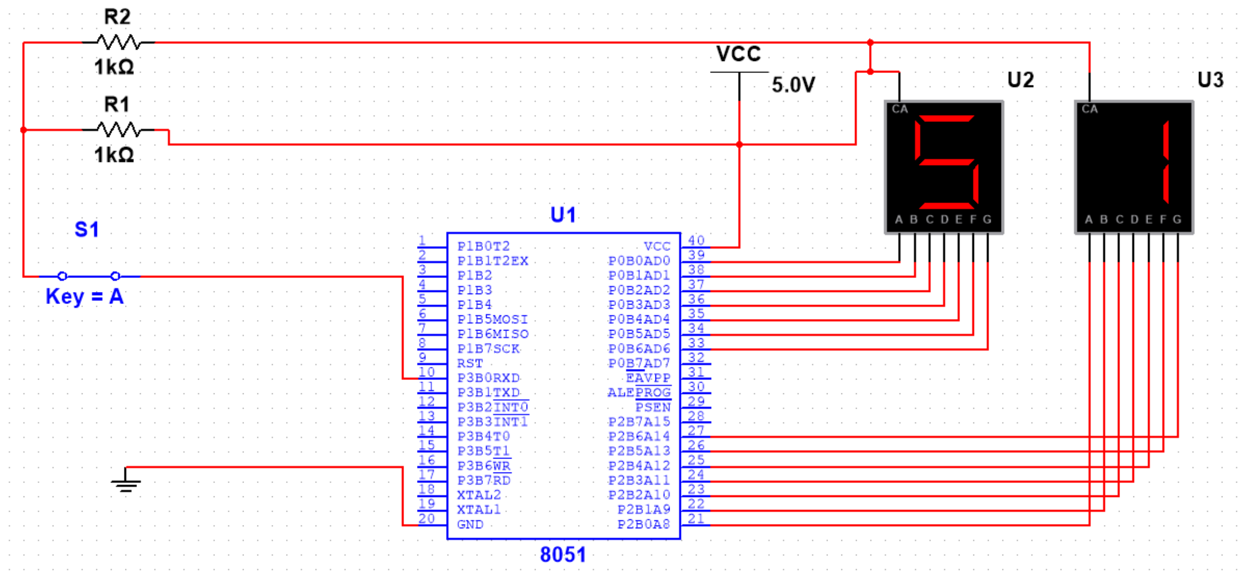
```
$MOD51
ORG 0000H
        mov P0,#0C0H
        acall DELAY
  MOV P0,#0F9H
        acall DELAY
  MOV P0,#0A4H
        acall DELAY
  MOV P0,#0B0H
        acall DELAY
  MOV P0,#99H
        acall DELAY
  MOV P0,#92H
        acall DELAY
  MOV P0,#82H
        acall DELAY
  MOV P0,#0F8H
        acall DELAY
  MOV P0,#80H
        acall DELAY
  MOV P0,#90H
        acall DELAY
DELAY:
  MOV R4, #80
```

```
DELAY_OUTER:
   MOV R3, #80
DELAY_INNER:
   DJNZ R3, DELAY_INNER
   DJNZ R4, DELAY_OUTER
   RET
END
```

C CODE:

```
#include <REG51.h>

void delay(unsigned int time) {
   unsigned int i, j;
   for(i = 0; i < time; i++)
      for(j = 0; j < 1275; j++);
}

void main() {
   while(1) {
      P0 = 0xC0; // Display '0'
      delay(1000);
      P0 = 0xF9; // Display '1'
      delay(1000);
      P0 = 0xA4; // Display '2'
      delay(1000);
      P0 = 0xB0; // Display '3'
      delay(1000);
      P0 = 0x99; // Display '4'
      delay(1000);
      P0 = 0x92; // Display '5'
      delay(1000);
      P0 = 0x82; // Display '6'
      delay(1000);
      P0 = 0xF8; // Display '7'
      delay(1000);
      P0 = 0x80; // Display '8'
      delay(1000);
      P2 = 0x90; // Display '9'
      delay(1000);
   }
}
```

QUESTION: 2



ASSEMBLY CODE:

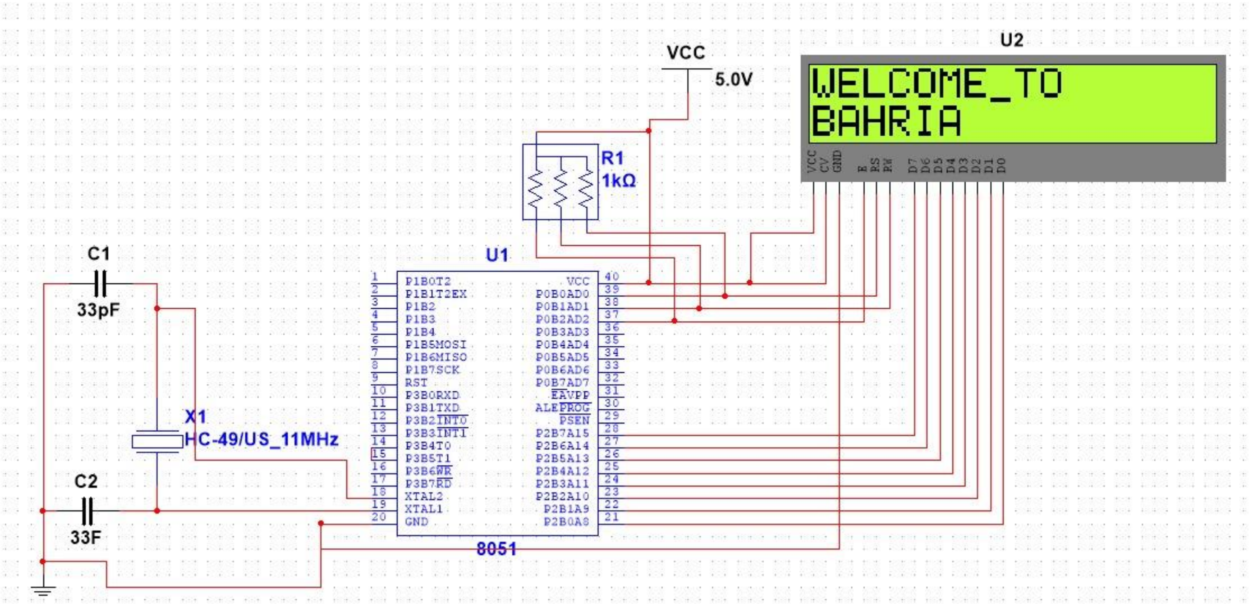
```
$MOD51
ORG 0000H
Main:
        acall seg21
        acall seg1
        acall seg22
        acall seg1
        acall seg23
        acall seg1
        acall seg24
        acall seg1
        acall seg25
        acall seg1
        acall seg26
        acall seg1
        acall seg27
        acall seg1
        acall seg28
        acall seg1
        acall seg29
seg1:
       MOV P2,#0C0H
       Acall DELAY
  MOV p2,#0F9H
       Acall DELAY
  MOV P2,#0A4H
       Acall DELAY
  MOV P2,#0B0H
```

```
        Acall DELAY
  MOV P2,#99H
        Acall DELAY
  MOV P2,#92H
        Acall DELAY
  MOV P2,#82H
        Acall DELAY
  MOV P2,#0F8H
        Acall DELAY
  MOV P2,#80H
        Acall DELAY
  MOV P2,#90H
        Acall DELAY
        ret
seg21:MOV P0,#0C0H
        acall DELAY
        ret
seg22:MOV P0,#0F9H
        acall DELAY
        ret
seg23:MOV P0,#0A4H
        acall DELAY
        ret
seg24:MOV P0,#0B0H
        acall DELAY
        ret
seg25:MOV P0,#99H
        acall DELAY
        ret
seg26:MOV P0,#92H
        acall DELAY
        ret
seg27:MOV P0,#82H
        acall DELAY
        ret
seg28:MOV P0,#0F8H
        acall DELAY
        ret
seg29: MOV P0,#90H
        acall DELAY
        ret
DELAY:
  MOV R4, #10
DELAY_OUTER:
  MOV R3, #55
DELAY_INNER:
  DJNZ R3, DELAY_INNER
  DJNZ R4, DELAY_OUTER
  RET
END
```

C CODE:

```
#include <REG51.h>
unsigned char segmentData[] = {
   0xC0,
   0xF9,
   0xA4,
   0xB0,
   0x99,
   0x92,
   0x82,
   0xF8,
   0x80,
   0x90
};

void delay(unsigned int time) {
   unsigned int i, j;
   for(i = 0; i < time; i++)
      for(j = 0; j < 1275; j++);
}

void main() {
   while(1) {
      P0 = segmentData[0];
      P2 = segmentData[1];
      delay(1000);
      P0 = segmentData[2];
      P2 = segmentData[3];
      delay(1000);
      P0 = segmentData[4];
      P2 = segmentData[5];
      delay(1000);
      P0 = segmentData[6];
      P2 = segmentData[7];
      delay(1000);
      P0 = segmentData[8];
      P2 = segmentData[9];
      delay(1000);
   }
}
```

QUESTION: 3



C CODE:

```
#include<reg51.h>
/*************** LCD control signals declaration *****************/
sbit RS = P0^0;
sbit RW = P0^1;  // Read/write line
sbit Enable = P0^2; // Enable line
#define LCD_PORT P2 // define port

/*************** LCD function prototypes ****************/
void send_cmd(unsigned char);
void send_char(unsigned char);
void LCD_init(void);
void delayms(unsigned int);
/*************** Main Funciton declaration *****************/
void main() {
LCD_PORT = 0x00; // Make the port as output port
LCD_init();          // LCD initialization
  send_cmd(0x80);     // Force cursor to beginning of 1st line, if the number is 0x83 then force the cursor
to 53rd position
  delayms(100);    // Delay of 100millisec
  send_char('W');  // Send data
  send_char('E');  // Send data
  send_char('L');  // Send data
  send_char('C');  // Send data
        send_char('O');  // Send data
```

```
            send_char('M');  // Send data
            send_char('E');  // Send data
            send_char(' ');  // Send data
            send_char('T');  // Send data
  send_char('O');  // Send data
  send_cmd(0xC0);     // Force cursor to beginning of 2nd line
  delayms(100);    // Delay of 100millisec
            send_char('B');  // Send data
  send_char('A');  // Send data
  send_char('H');  // Send data
            send_char('R');  // Send data
            send_char('I');  // Send data
            send_char('A');  // Send data
}

/*************** LCD Initialization Function declaration ************/
void LCD_init()   {
 send_cmd(0x38);      // configuring LCD as 2 line 5x7 matrix
 send_cmd(0x0E);      // Display on, Cursor blinking
 send_cmd(0x01);      // Clear Display Screen
 send_cmd(0x06);      // Increment Cursor (Right side)
}

void send_char(unsigned char character) {
 LCD_PORT = character;
 RS = 1;    // Select Data Register
 RW = 0;    // write operation
 Enable = 1;      // High to Low pulse provided on the enable pin with nearly 1ms(>450ns)
 delayms(1);   // 1 millisec delay
 Enable = 0;
 delayms(1);   // 100 millisec delay
}

/***************LCD Command Sending Function declaration***********/
void send_cmd(unsigned char Command) {
 LCD_PORT = Command;
 RS = 0;      // Select Command Register
 RW = 0;    // write operation
 Enable = 1;      // High to Low pulse provided on the enable pin with nearly 1ms(>450ns)
 delayms(1);   // 1 millisec delay
 Enable = 0;
}

/*************** delayms Function declaration****************/
```

```
void delayms(unsigned int val) {
 unsigned int i,j;
 for(i=0;i<=val;i++) {
  for(j=0;j<=2;j++);
  //nop();   // no operation produce 1us time delay
 }
}
//lcd2.c;
```

ASSEMBLY CODE:

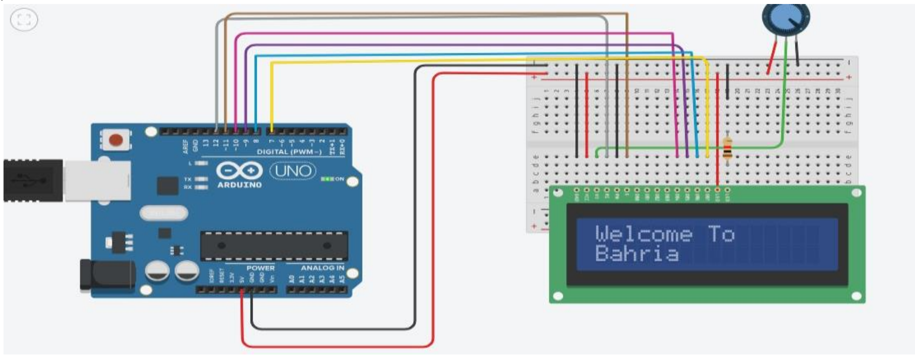
```
; Define control signals for LCD
#define RS P3.0
#define RW P3.1
#define EN P3.2

; Define LCD data port
LCD_data EQU P2

ORG 0x0000
   MOV  TMOD, #0x20   ; Timer 1, mode 2
   MOV  TH1, #0xFD    ; Set baud rate to 9600
   SETB TR1           ; Start Timer 1
   MOV  SCON, #0x50   ; Set serial port mode

   MOV  A, #0x38      ; Initialize the LCD
   CALL LCD_Init

   MOV  DPTR, #Welcome
   CALL LCD_DisplayString

   SJMP $

; Initialize the LCD
LCD_Init:
   MOV  RS, #0
   MOV  RW, #0
   MOV  A, #0x38      ; Function set: 2 lines, 5x7 matrix
   CALL LCD_Cmd
   MOV  A, #0x0E      ; Display on, cursor on, blink on
   CALL LCD_Cmd
   MOV  A, #0x01      ; Clear display
   CALL LCD_Cmd
   MOV  A, #0x06      ; Entry mode set: increment, no display shift
   CALL LCD_Cmd
   RET
```

```
; Send command to the LCD
LCD_Cmd:
   MOV  LCD_data, A
   CLR  RS
   CLR  RW
   SETB EN
   CLR  EN
   RET

; Display a string on the LCD
LCD_DisplayString:
   MOV  A, #0x80      ; Set cursor to the first line
   CALL LCD_Cmd
Loop:
   MOV  A, @DPTR
   CJNE A, #0, Continue
   RET
Continue:
   CALL LCD_Data
   INC  DPTR
   SJMP Loop

Welcome: DB "WELCOME_TO BAHRIA", 0

END
```

LCD Interface using Arduino

PROGRAM:

```
#include<LiquidCrystal.h>
LiquidCrystal lcd(12,11,10,9,8,7);
 void setup()
{
  lcd.begin(16,2);
}
   void loop()
   {
     lcd.setCursor(0,0);
     lcd.print("Welcome To");
     lcd.setCursor(0,1);
     lcd.print("Bahria");
   }
```


UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN
Welcome To
Bahria