

## EMBEDDED SYSTEMS-II

(Programs for Record)

~ VBS\_402

1).Write an embedded C program to blink the 1 LED . [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <1 LED Blink.h>
```

```
#bit LED = 0X06.1
```

```
#bit TLED = 0X86.1
```

```
void main()
```

```
{
```

```
TLED = 0;
```

```
LED = 0;
```

```
while(TRUE)
```

```
{
```

```
LED = 1;
```

```
delay_ms(1000);
```

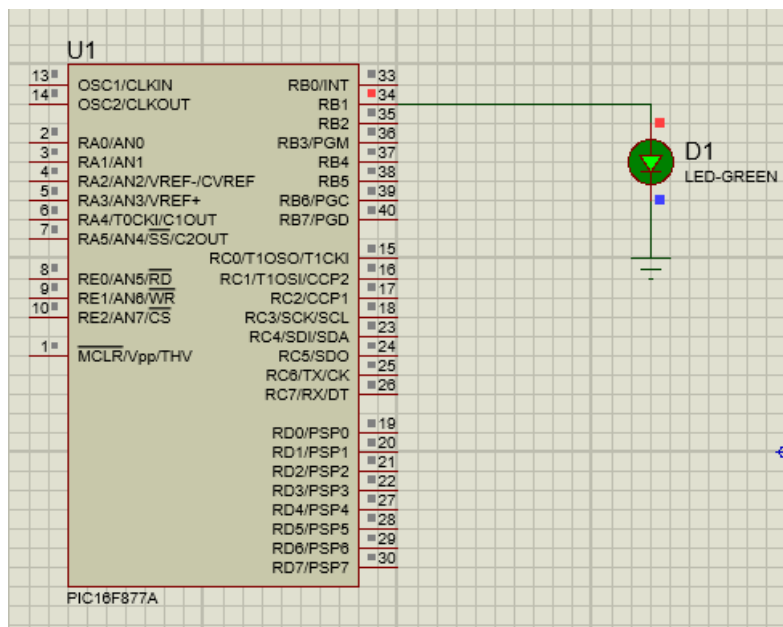
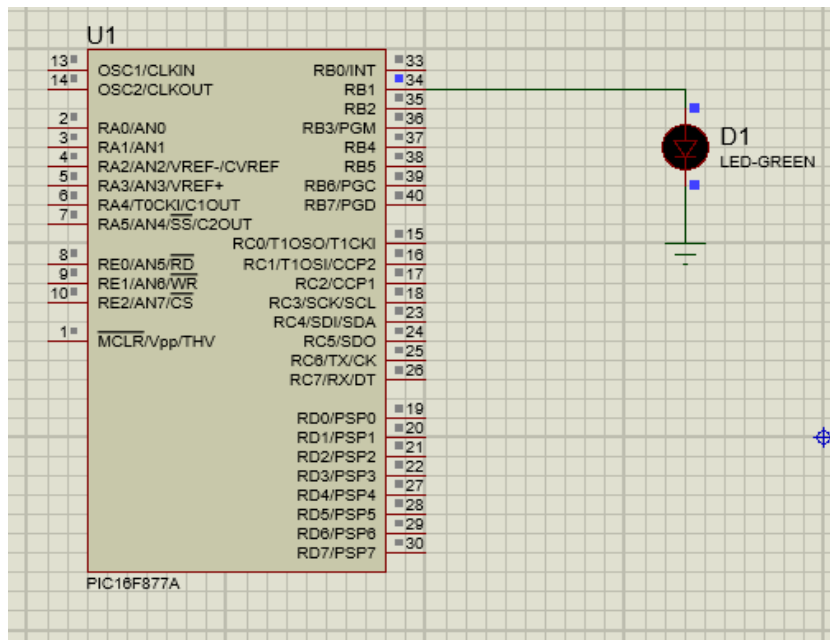
```
LED=0;
```

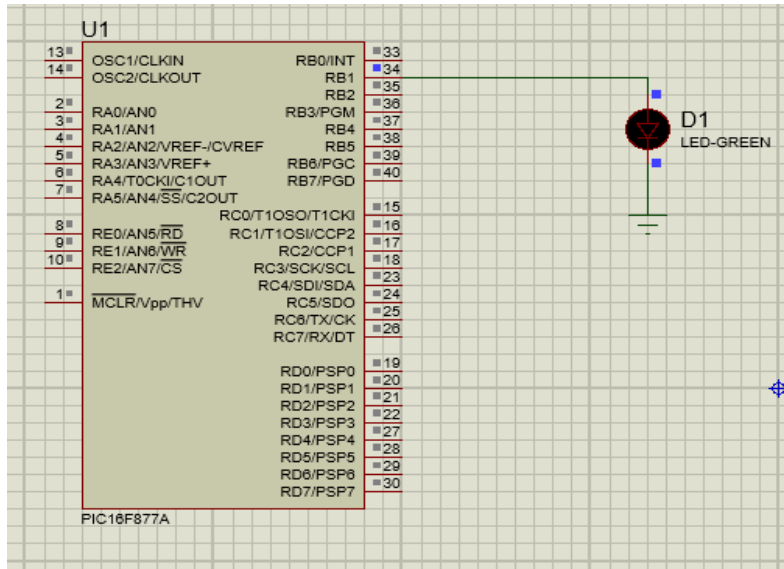
```
delay_ms(1000);
```

```
}
```

```
}
```

**SIMULATION OUTPUT:**





2). Write an embedded C program to blink the 1 LED . [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <2 LED Blink.h>
```

```
#bit LED1=0X06.1
```

```
#bit TLED1=0X86.1
```

```
#bit LED2=0X07.1
```

```
#bit TLED2=0X87.1
```

```
void main()
```

```
{
```

```
TLED1=0;
```

```
LED1=0;
```

```
TLED2=0;
```

```
LED2=0;
```

```
while(TRUE)
```

```
{
```

```
LED1=1;
```

```
LED2=1;
```

```
delay_ms(1000);
```

```
LED1=0;
```

```

LED2=0;

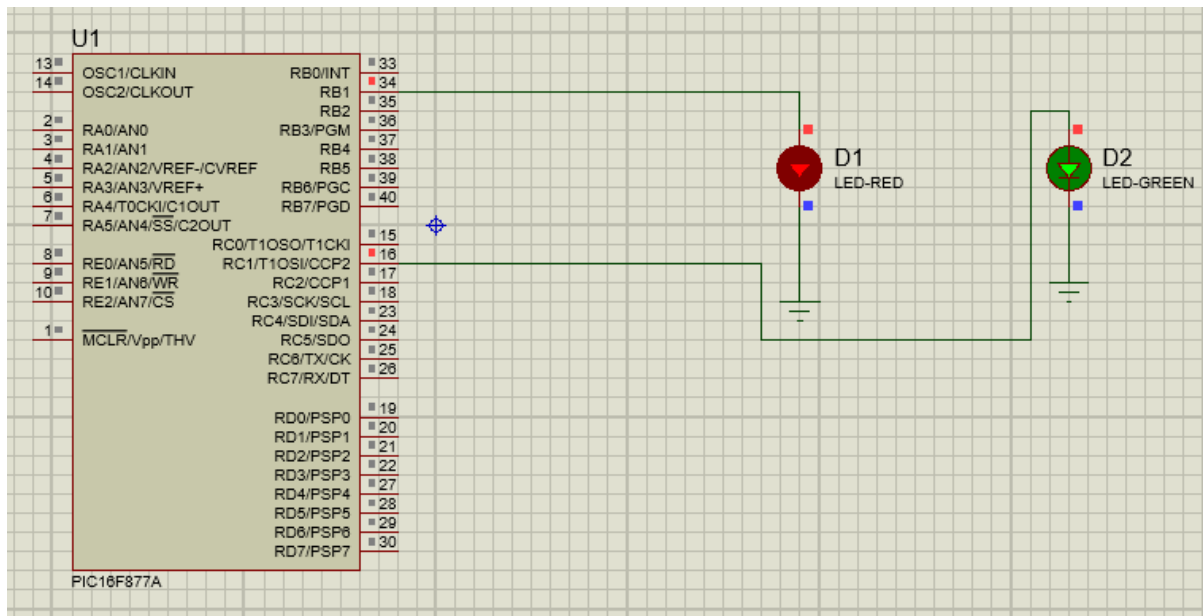
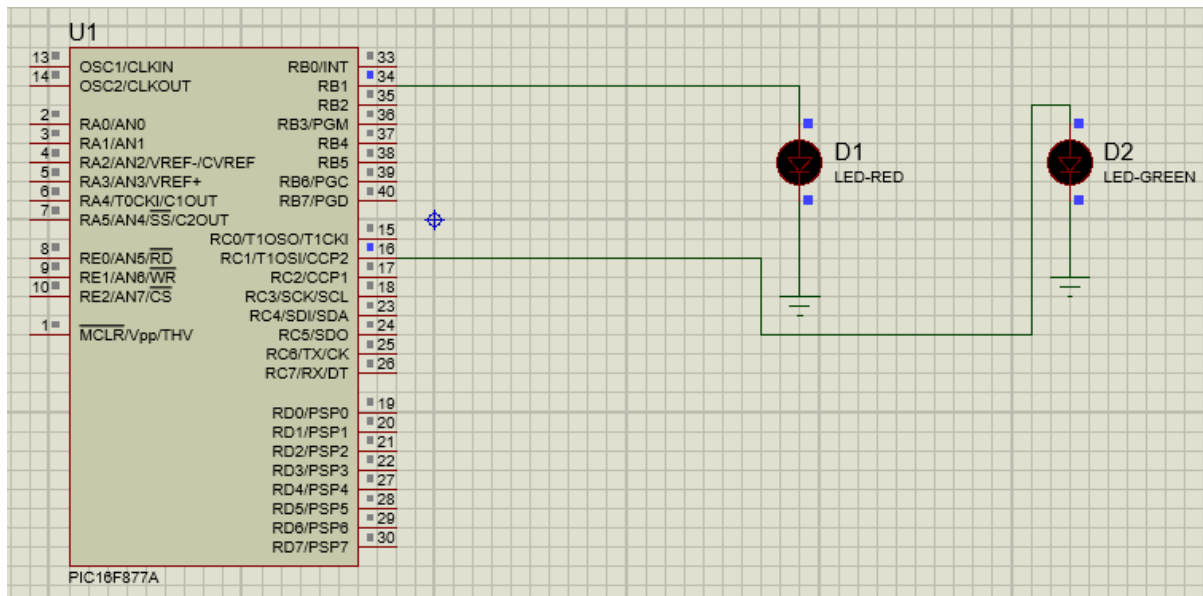
delay_ms(1000);

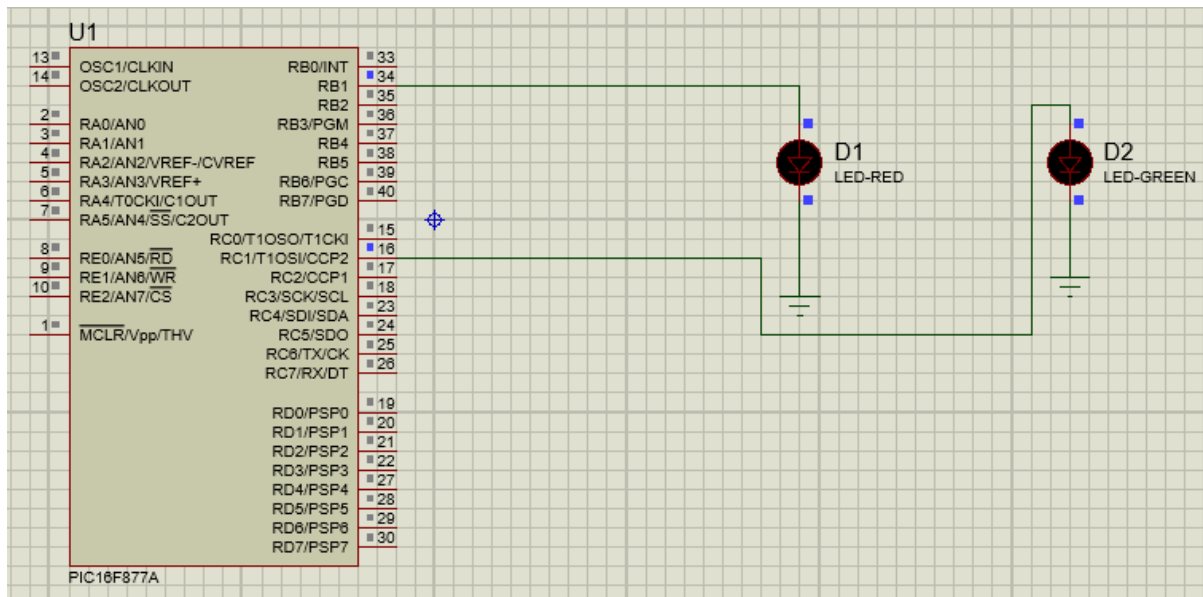
}

}

```

### SIMULATION OUTPUT:





3). Write an embedded C program to blink the even LEDs one time & then some delay & then ODD LED blink & then some delay . [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <8 LED even odd.h>
```

```
#bit LED1=0X06.0
```

```
#bit TLED1=0x86.0
```

```
#bit LED2=0X06.1
```

```
#bit TLED2=0X86.1
```

```
void main()
```

```
{
```

```
LED1=0;
```

```
TLED1=0;
```

```
LED2=0;
```

```
TLED2=0;
```

```
while(TRUE)
```

```
{
```

```
LED1=1;
```

```
delay_ms(1000);
```

```

LED1=0;

delay_ms(1000);

LED2=1;

delay_ms(1000);

LED2=0;

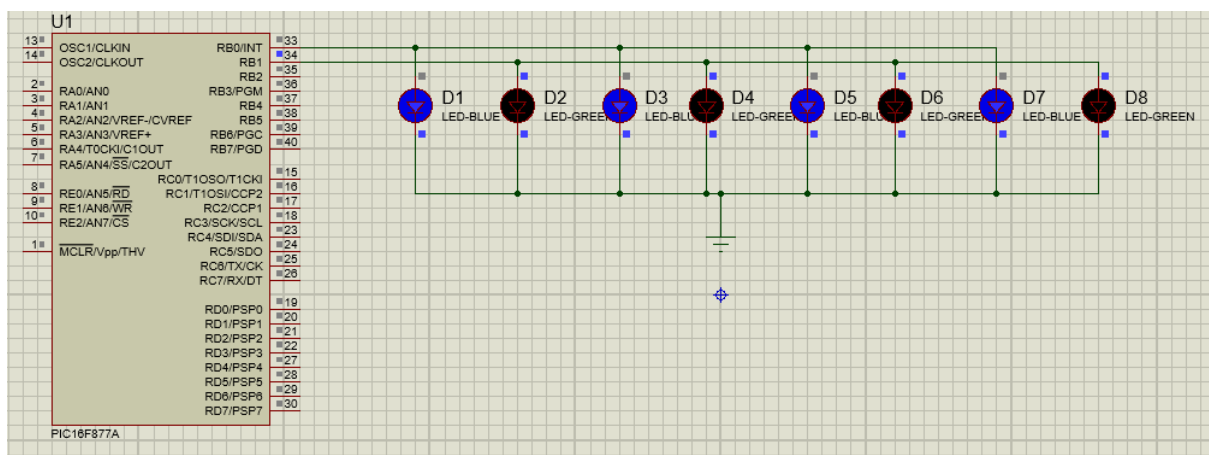
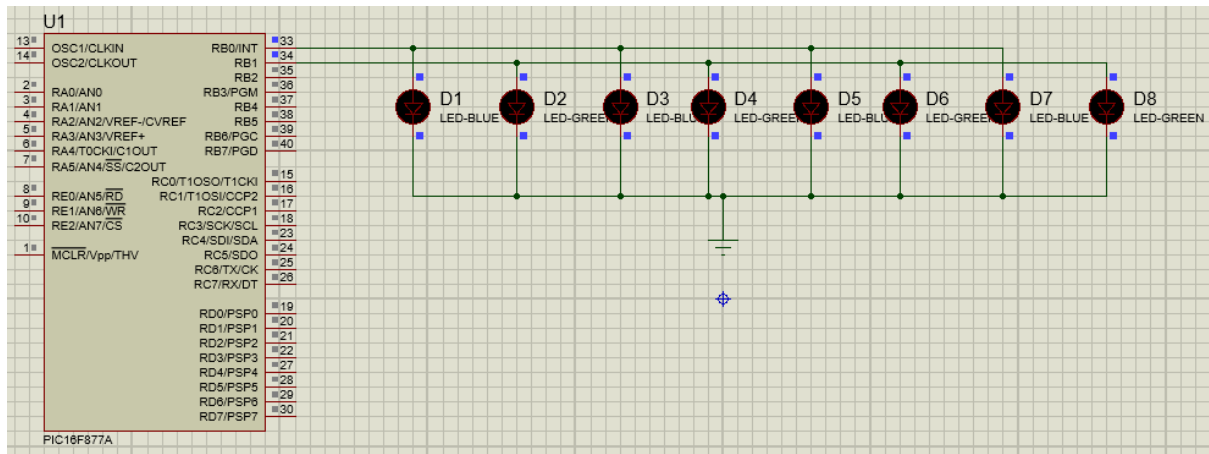
delay_ms(1000);

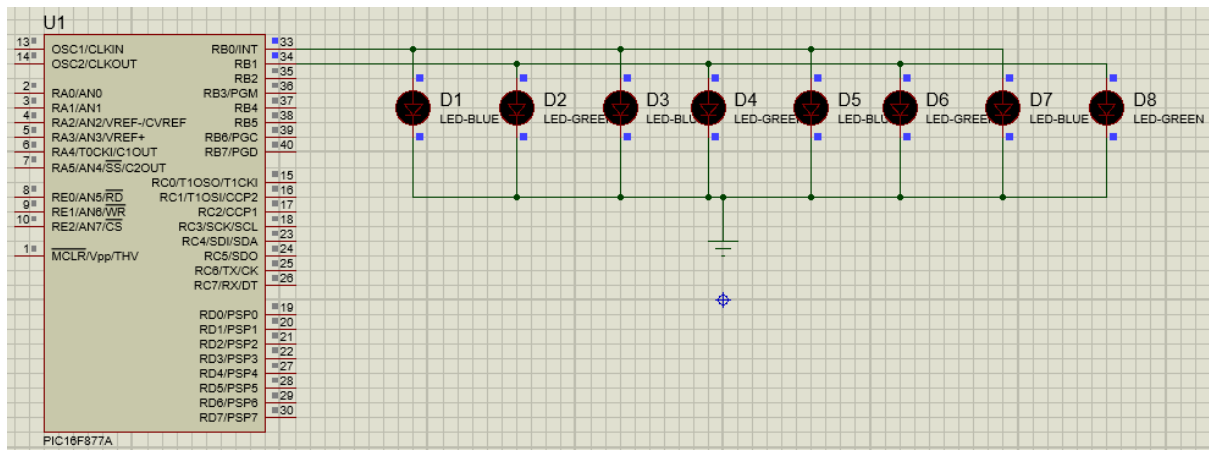
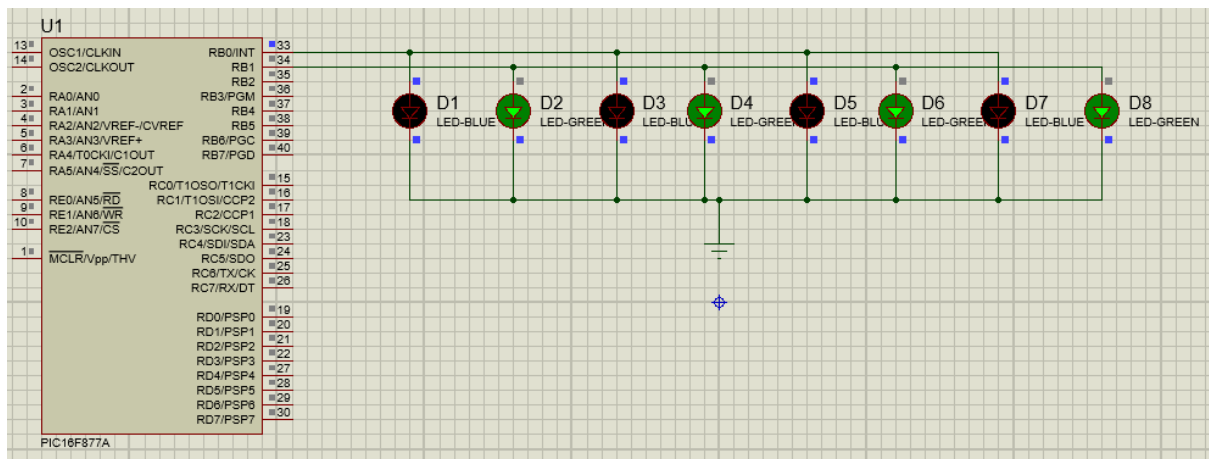
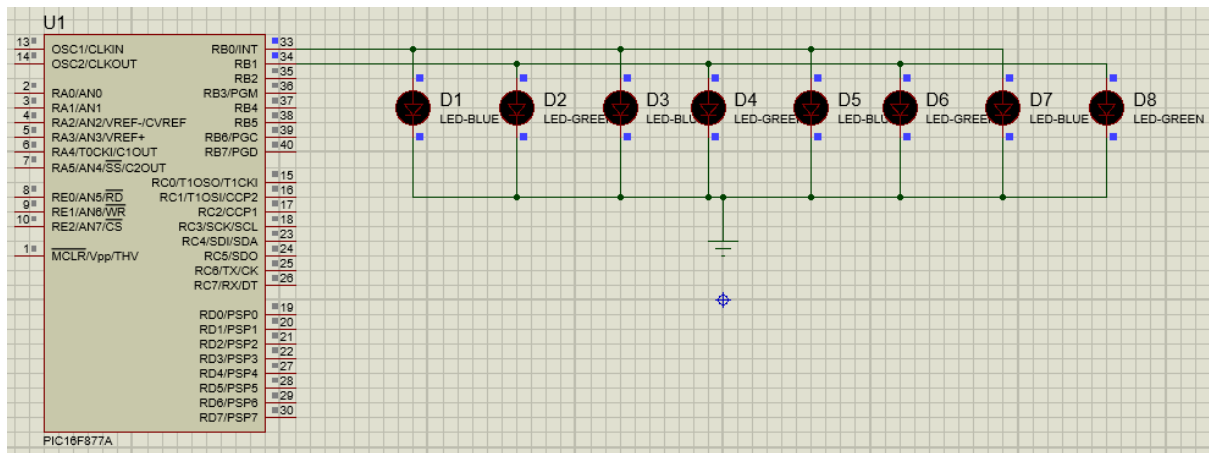
}

}

```

### SIMULATION OUTPUT:





4). Write an embedded C program to blink 1 LED using 1 SWITCH . [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <1 SWITCH 1 LED.h>
```

```
#bit sw=0X06.1
```

```
#bit tsw=0x86.1
```

```
#bit led=0x07.0
```

```
#bit tled=0x87.0
```

```
void main()
```

```
{
```

```
sw=0;
```

```
tsw=1;
```

```
led=0;
```

```
tled=0;
```

```
while(TRUE)
```

```
{
```

```
if(sw==1)
```

```
{
```

```
led=1;
```

```
}
```

```
else
```

```
{
```

```
led=0;
```

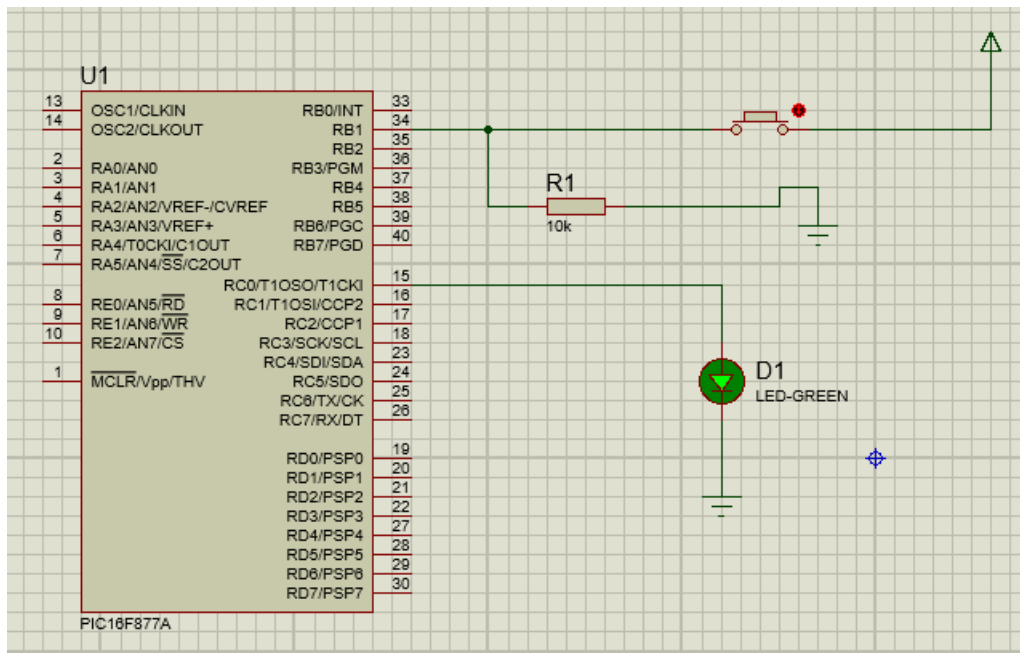
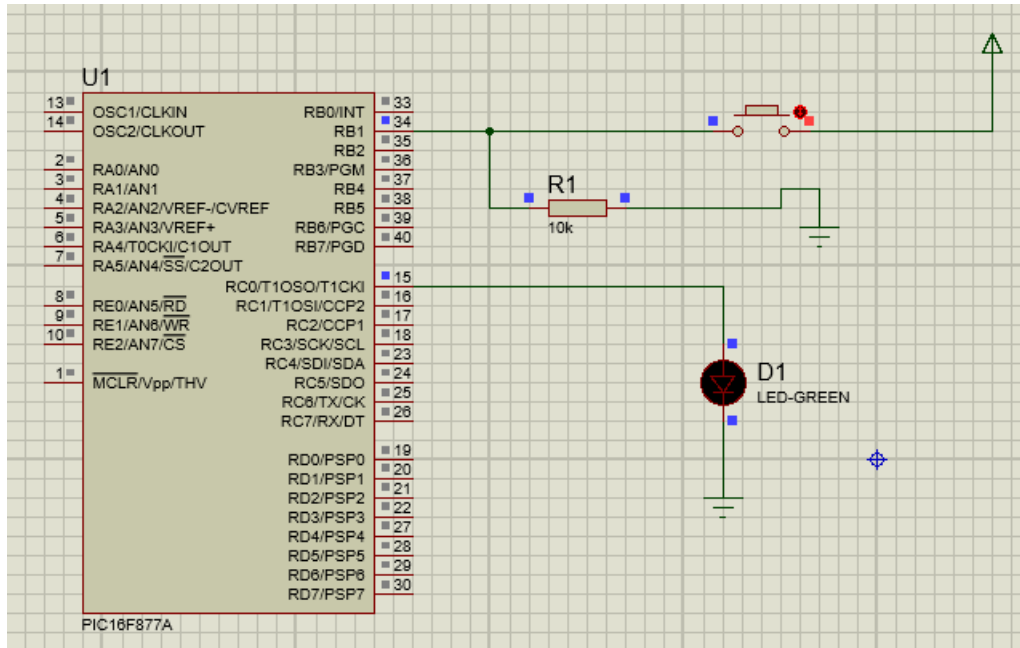
```
}
```

```
}
```

```
}
```

**SIMULATION OUTPUT:**





5). Write an embedded C program to blink 1 LED using 2 SWITCH,when one switch is pressed LED must turn ON & another switch is pressed Led must turn OFF. [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <2 SWITCH 1 LED.h>
```

```
#bit sw1=0X06.0
```

```
#bit tsw1=0X86.0
```

```
#bit sw2=0x06.1
```

```
#bit tsw2=0X86.1
```

```
#bit LED=0X07.0
```

```
#bit TLED=0X87.0
```

```
void main()
```

```
{
```

```
sw1=0;
```

```
tsw1=1;
```

```
sw2=0;
```

```
tsw2=1;
```

```
LED=0;
```

```
TLED=0;
```

```
while(TRUE)
```

```
{
```

```
if(sw1==1)
```

```
{
```

```
LED=1;
```

```
}
```

```
if(sw2==1)
```

```
{
```

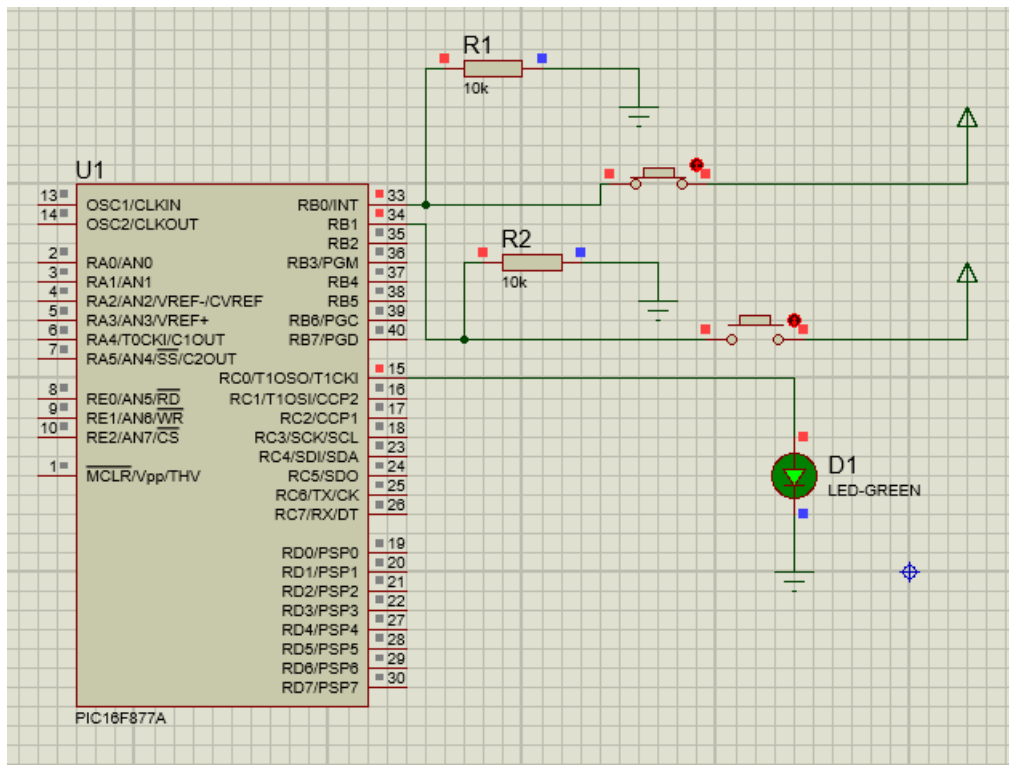
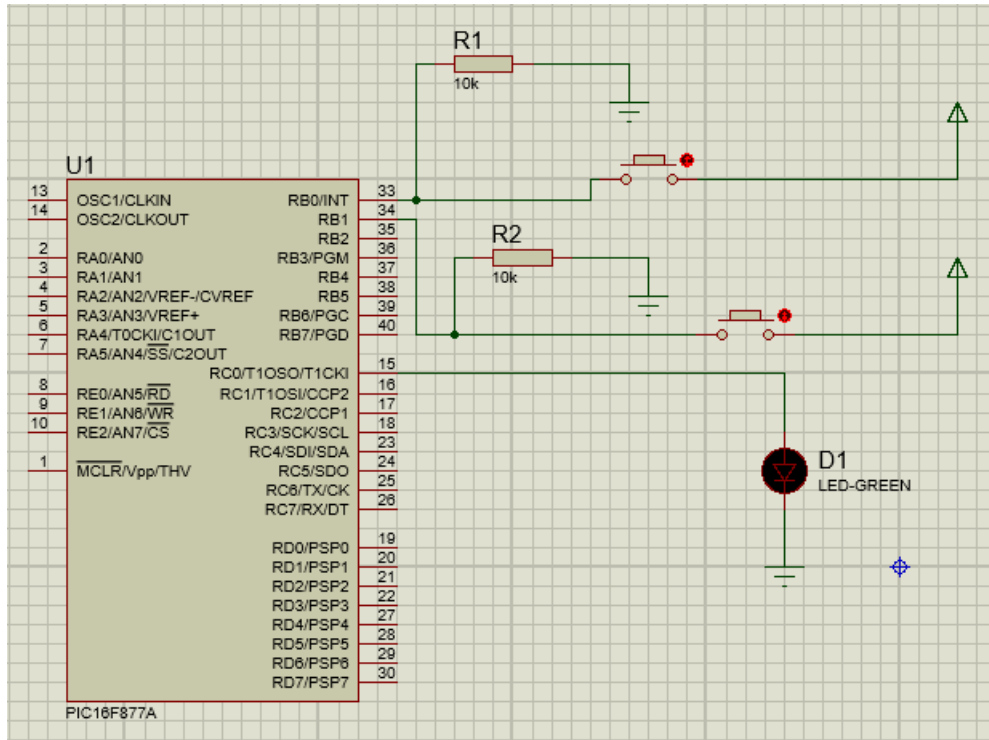
```
LED=0;
```

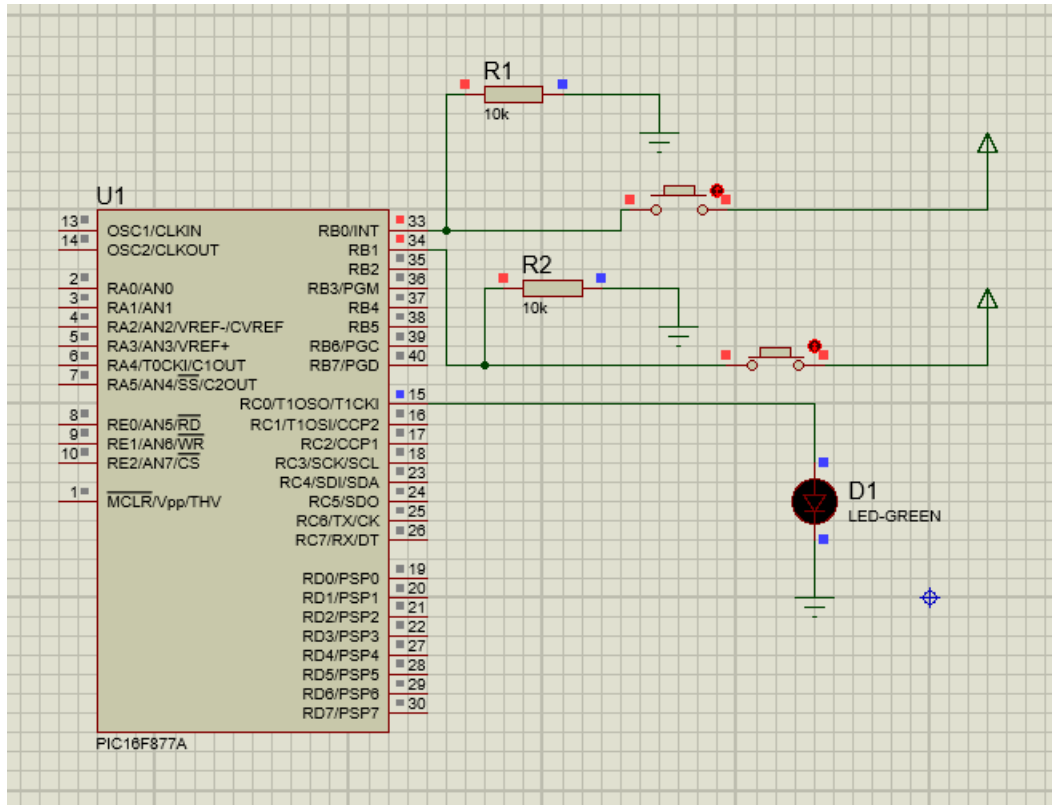
```
}
```

```
}
```

```
}
```

**SIMULATION OUTPUT:**





6). Write an embedded C program to turn the LED ON when only both switches are ON condition. [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <2 ON SWITCH 1 LED.h>
```

```
#bit sw1=0x06.0
```

```
#bit tsw1=0x86.0
```

```
#bit sw2=0x06.1
```

```
#bit tsw2=0x86.1
```

```
#bit LED=0x07.0
```

```
#bit TLED=0x87.0
```

```
void main()
```

```
{
```

```
sw1=0;
```

```
tsw1=1;
```

```

sw2=0;

tsw2=1;

LED=0;

TLED=0;

while(TRUE)

{

if(sw1==1 && sw2==1)

{

LED=1;

}

else

{

LED=0;

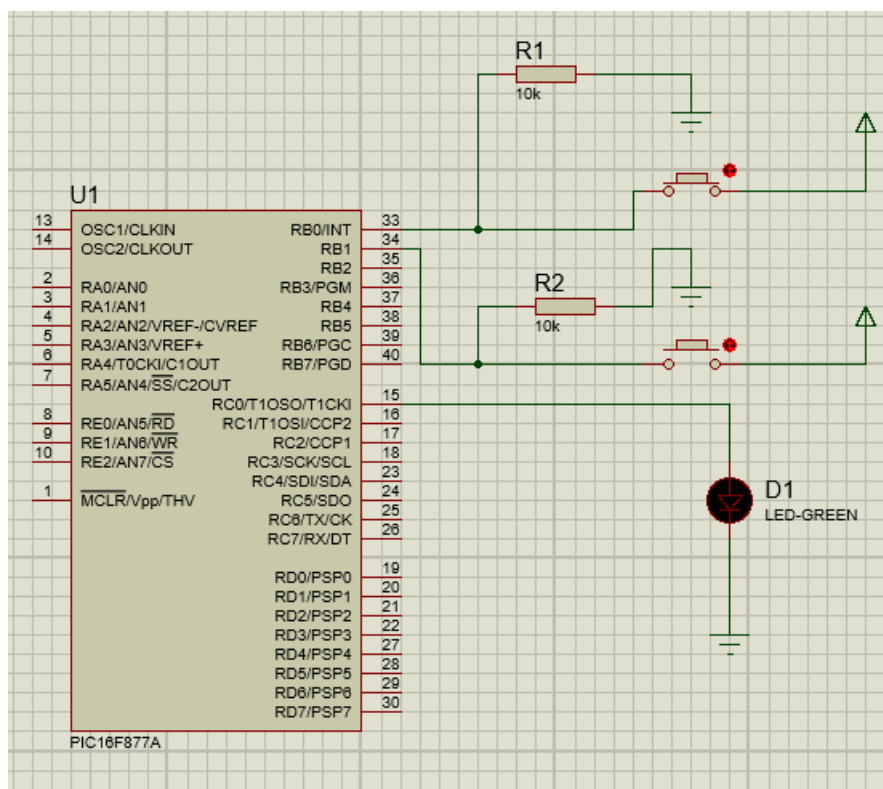
}

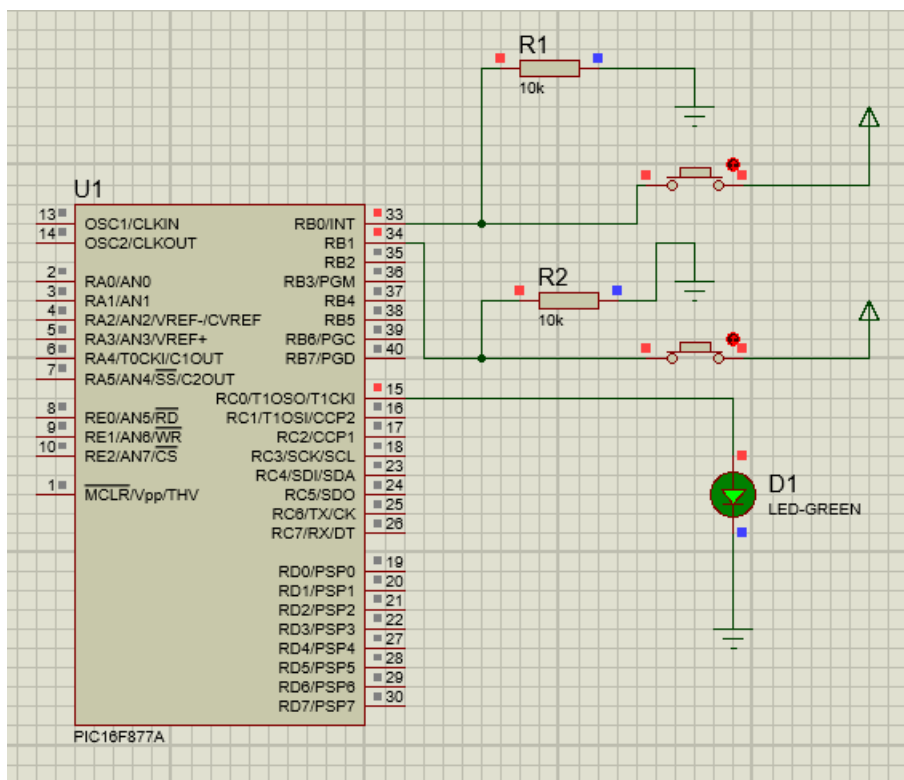
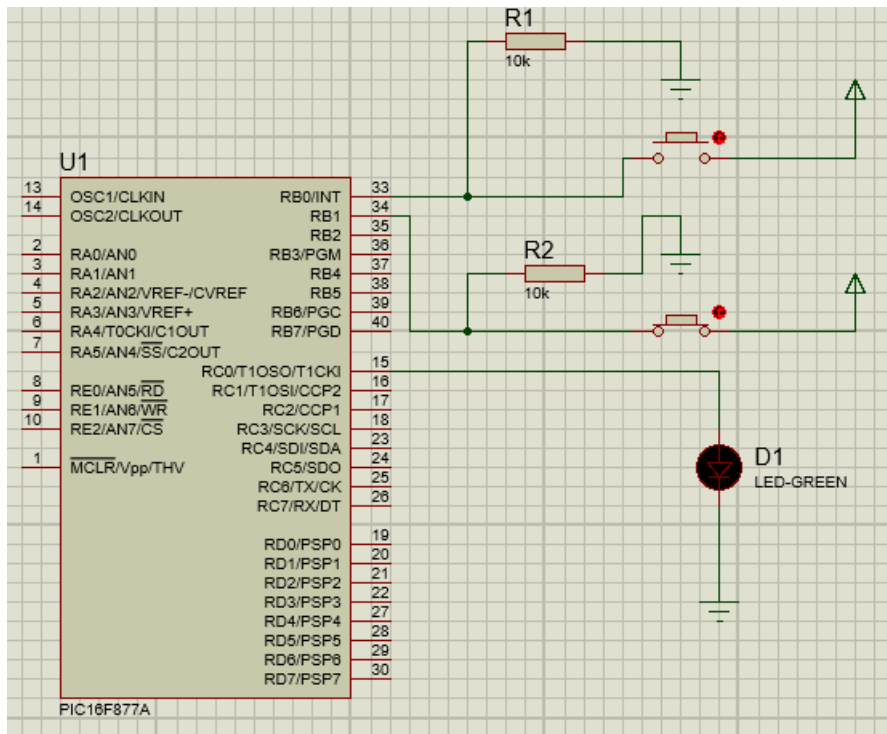
}

}

```

### SIMULATION OUTPUT:





7). Write an embedded C program to display a Character on LCD. [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <1 LCD DISPLAY A.h>
```

```

#byte LCD=0X08

#byte TLCD=0X88

#bit RS=0X07.0

#bit TRS=0X87.0

#bit EN=0X07.1

#bit TEN=0X87.1

void display(unsigned int A,unsigned int B)
{
    LCD=A;

    RS=B;

    EN=1;

    delay_ms(200);

    EN=0;
}

void main()
{
    LCD=0;

    TLCD=0;

    RS=0;

    TRS=0;

    EN=0;

    TEN=0;

    display(0X0C,0);

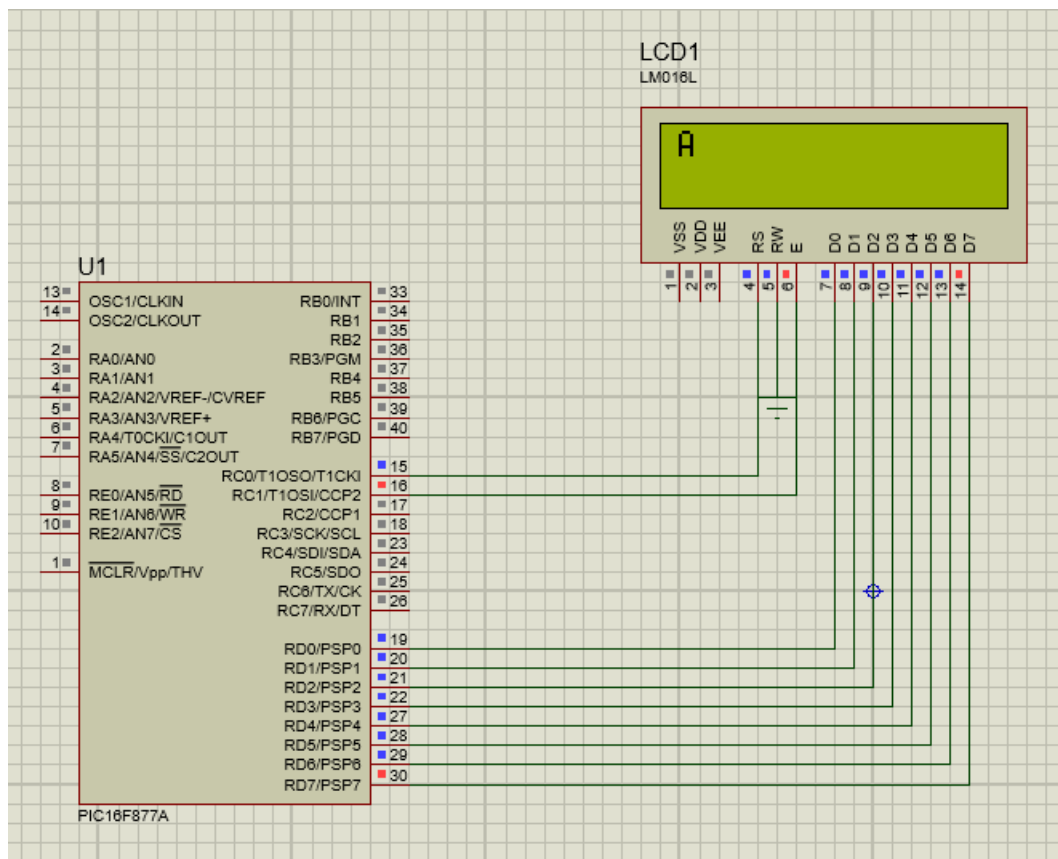
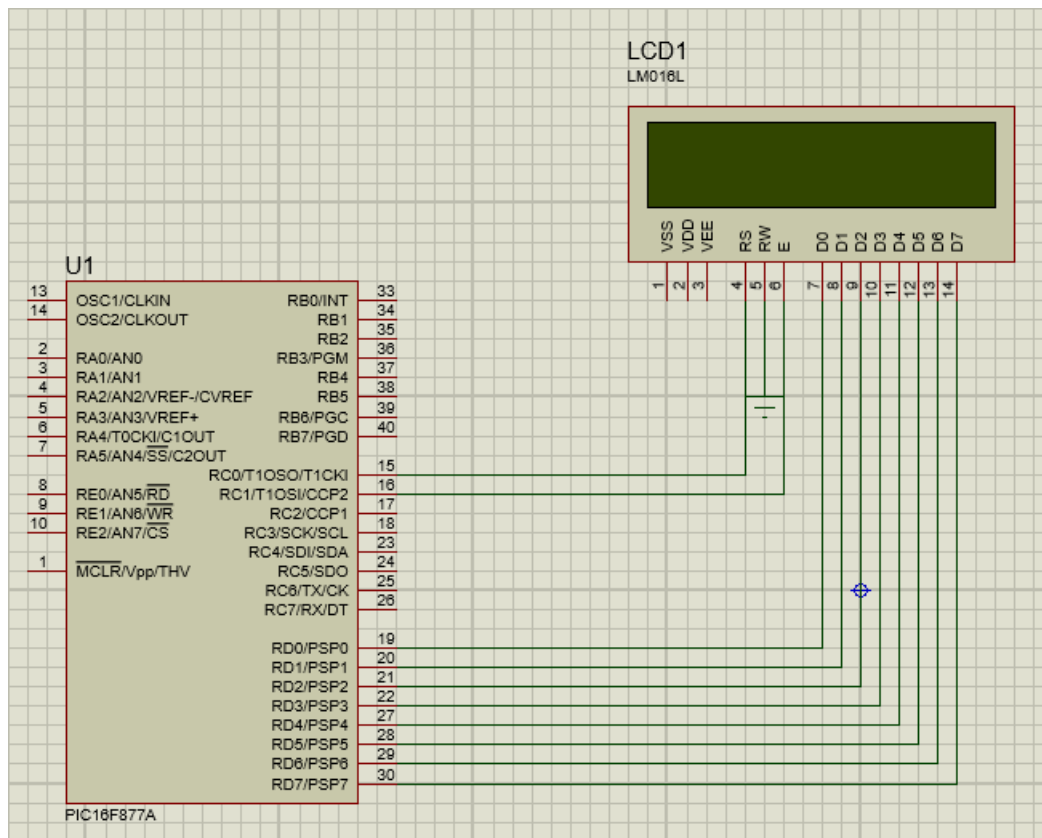
    display(0X38,0);

    while(TRUE)
    {
        display(0X80,0);

        display('A',1);
    }
}

```

## SIMULATION OUTPUT:





8). Write an embedded C program to display a Word/name/QISCET on LCD. [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <1 LCD DISPLAY QISCET & ECE.h>
```

```
#byte LCD=0X08
```

```
#byte TLCD=0X88
```

```
#bit RS=0X07.0
```

```
#bit TRS=0X87.0
```

```
#bit EN=0X07.1
```

```
#bit TEN=0X87.1
```

```
void display(unsigned int A,unsigned int B)
```

```
{
```

```
LCD=A;
```

```
RS=B;
```

```
EN=1;
```

```
delay_ms(200);
```

```
EN=0;
```

```
}
```

```
void main()
```

```
{
```

```
LCD=0;
```

```
TLCD=0;
```

```
RS=0;
```

```
TRS=0;
```

```
EN=0;
```

```
TEN=0;
```

```
display(0X0C,0);
```

```
display(0X38,0);
```

```
while(TRUE)
```

```
{
```

```
display(0X80,0);
```

```

display('Q',1);

display(0X81,0);

display('I',1);

display(0X82,0);

display('S',1);

display(0X83,0);

display('C',1);

display(0X84,0);

display('E',1);

display(0X85,0);

display('T',1);

display(0XC0,0);

display('E',1);

display(0XC1,0);

display('C',1);

display(0XC2,0);

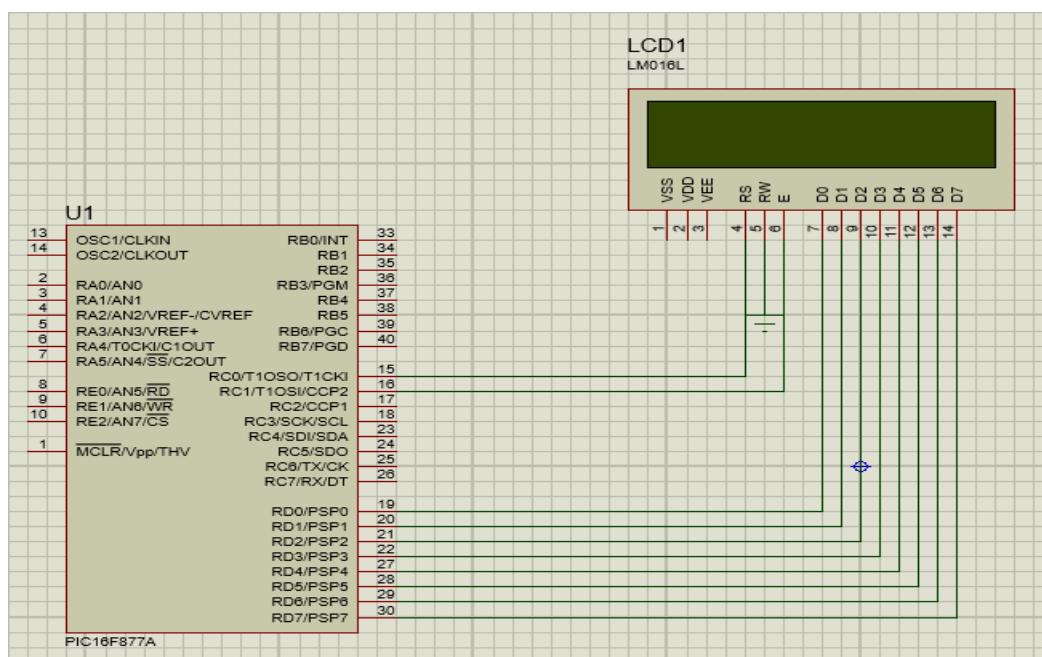
display('E',1);

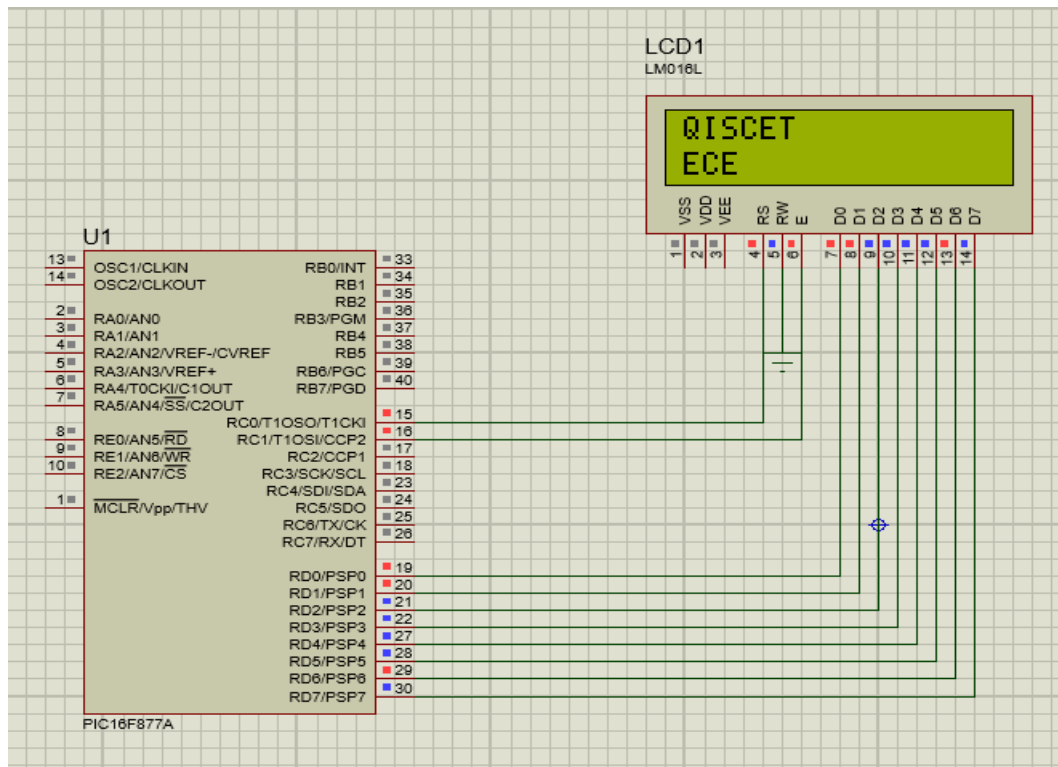
}

}

```

### SIMULATION OUTPUT:





9). Write an embedded C program to display a Word/name/QISCET on LCD using Strings. [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <1 LCD DISPLAY A.h>
```

```
#byte LCD=0X08
```

```
#byte TLCD=0X88
```

```
#bit RS=0X07.0
```

```
#bit TRS=0X87.0
```

```
#bit EN=0X07.1
```

```
#bit TEN=0X87.1
```

```
char STR[8]="QISCET";
```

```
void display(unsigned int A,unsigned int B)
```

```
{
```

```
LCD=A;
```

```
RS=B;
```

```
EN=1;
```

```
delay_ms(200);
```

```

EN=0;

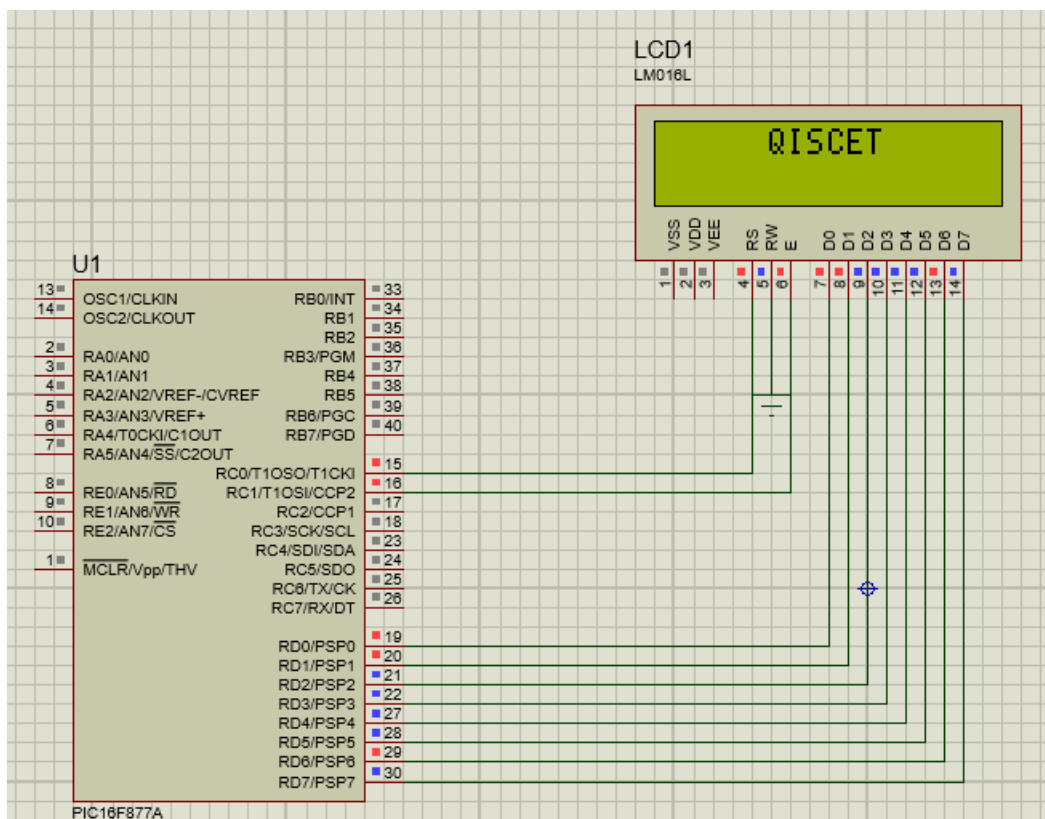
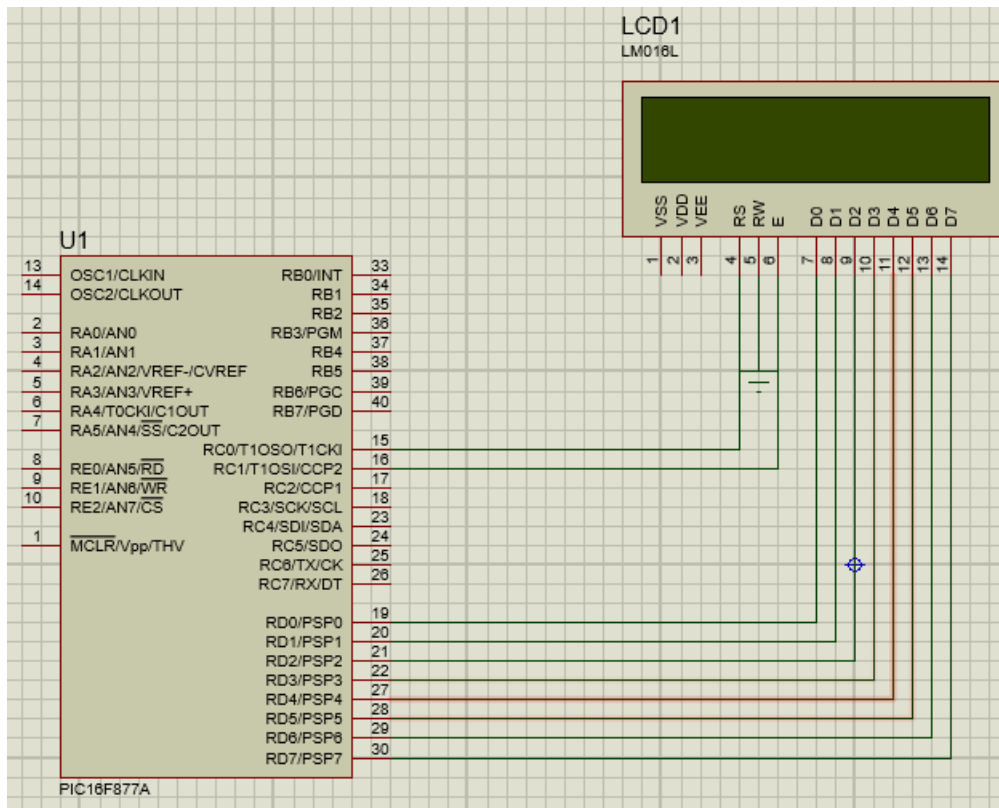
}

void display_string(char *str)
{
for(int i=0;str[i]!=0;i++)
{
display(str[i],1);
delay_ms(200);
}
}

void main()
{
LCD=0;
TLCD=0;
RS=0;
TRS=0;
EN=0;
TEN=0;
display(0X0C,0);
display(0X38,0);
while(TRUE)
{
display(0X85,0);
display_string(STR);
}
}

```

**SIMULATION OUTPUT:**



10). Write an embedded C program to turn the LEDS ,LED-1 will be turned ON when COUNT>=0, LED-2 will be turned ON when COUNT>=5, LED-3 will be turned ON when COUNT>=8, LED-4 will be turned ON when COUNT>=10, using Two Switches for increamenting and decreamenting the COUNT. [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <SENSOR TYPE 3.h>
```

```
#byte LCD=0x08
```

```
#byte TLCD=0x88
```

```
#bit RS=0x07.6
```

```
#bit TRS=0x87.6
```

```
#bit EN=0x07.7
```

```
#bit TEN=0x87.7
```

```
#bit SW1=0x07.0
```

```
#bit TSW1=0x87.0
```

```
#bit SW2=0x07.1
```

```
#bit TSW2=0x87.1
```

```
#bit LED1=0x06.0
```

```
#bit TLED1=0x86.0
```

```
#bit LED2=0x06.1
```

```
#bit TLED2=0x86.1
```

```
#bit LED3=0x06.2
```

```
#bit TLED3=0x86.2
```

```
#bit LED4=0x06.3
```

```
#bit TLED4=0x86.3
```

```
char str[10]="COUNT : ";
```

```
int count=0;

VOID display(unsigned int A,unsigned int B)
{
LCD=A;

RS=B;

EN=1;

delay_ms(200);

EN=0;

}

void display_string(char *str)
{
for(int j=0;str[j]!=0;j++)
{
display(str[j],1);

delay_ms(200);

}

}

void main()
{

SW1=0;

TSW1=1;

SW2=0;

TSW2=1;

LCD=0;

TLCD=0;

RS=0;

TRS=0;

EN=0;

TEN=0;


LED1=0;
```

```
TLED1=0;
```

```
LED2=0;
```

```
TLED2=0;
```

```
LED3=0;
```

```
TLED3=0;
```

```
LED4=0;
```

```
TLED4=0;
```

```
display(0x0c,0);
```

```
display(0x38,0);
```

```
while(TRUE)
```

```
{
```

```
    display(0x80,0);
```

```
    display_string(str);
```

```
    if(SW1==1)
```

```
    {
```

```
        while(SW1==1);
```

```
        count+=1;
```

```
    }
```

```
    if(SW2==1)
```

```
    {
```

```
        while(SW2==1);
```

```
        count-=1;
```

```
    }
```

```
    display(0x89,0);
```

```
    display((count/10)+0x30,1);
```

```
    display(0x8A,0);
```

```
    display((count%10)+0x30,1);
```

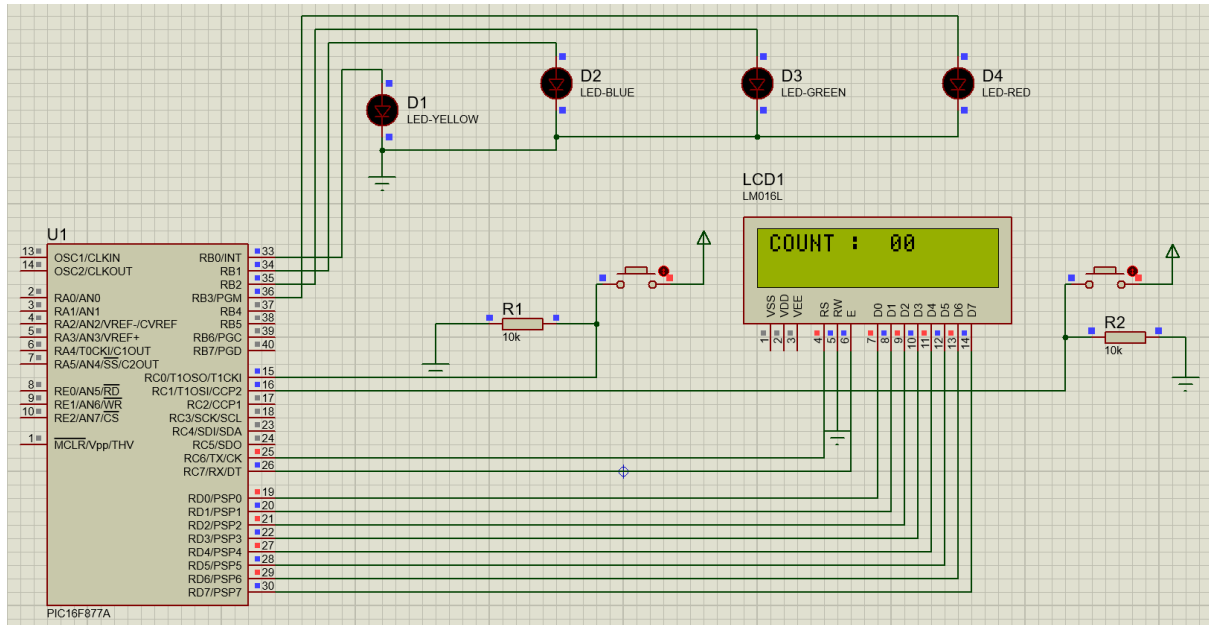
```
    if(count>=1)
```

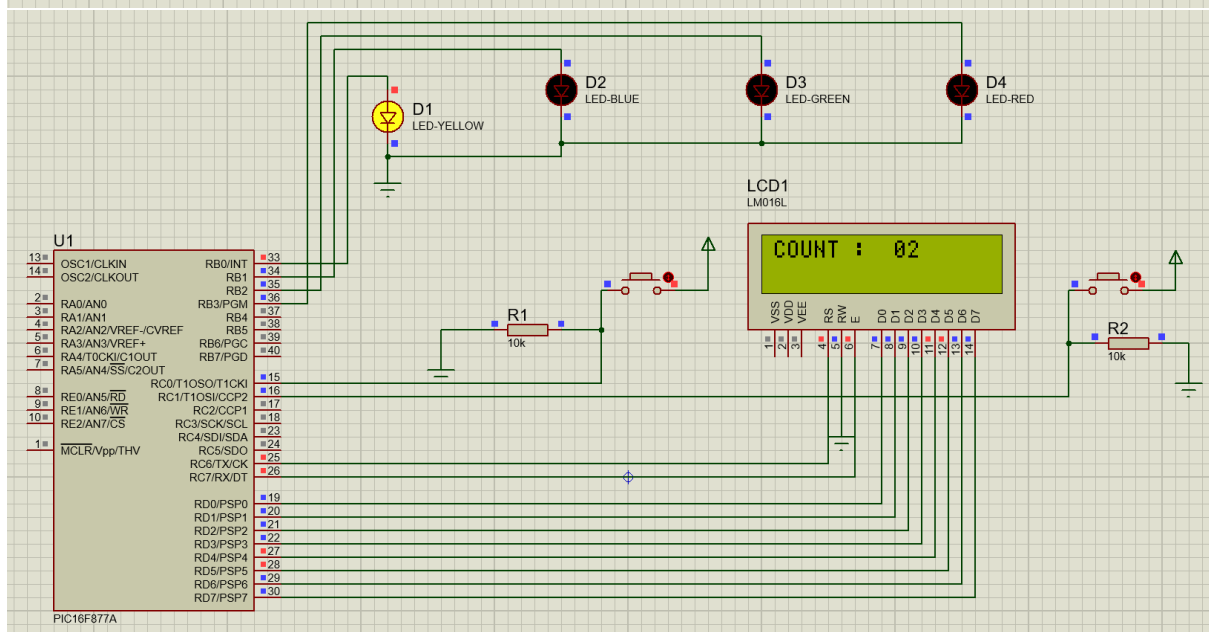
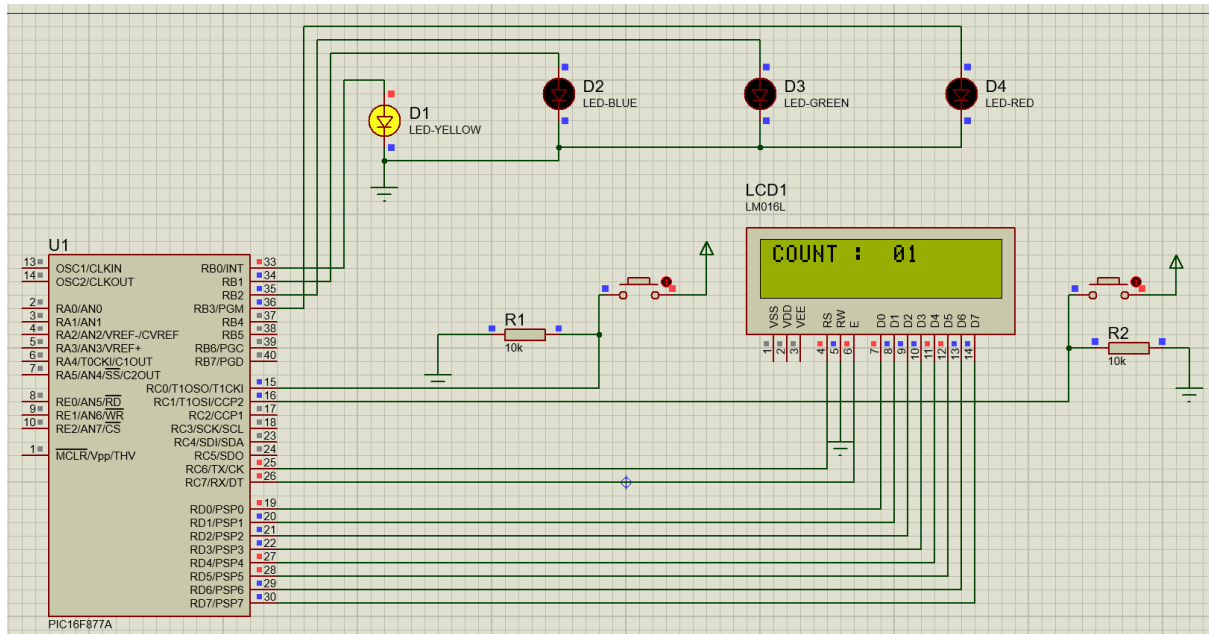


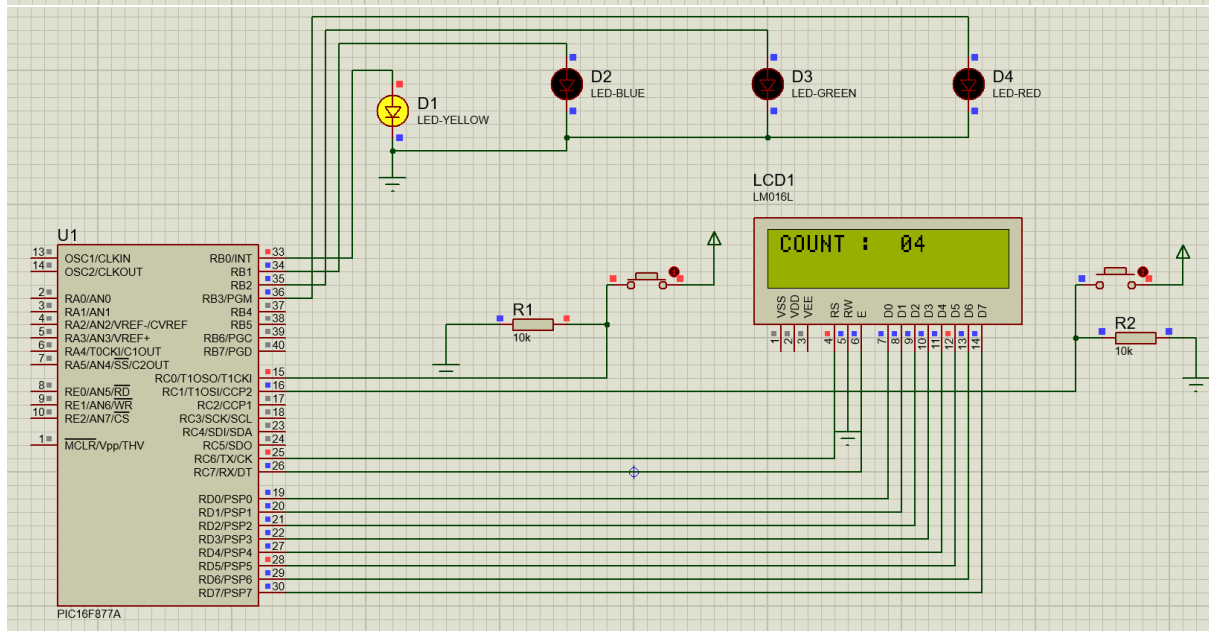
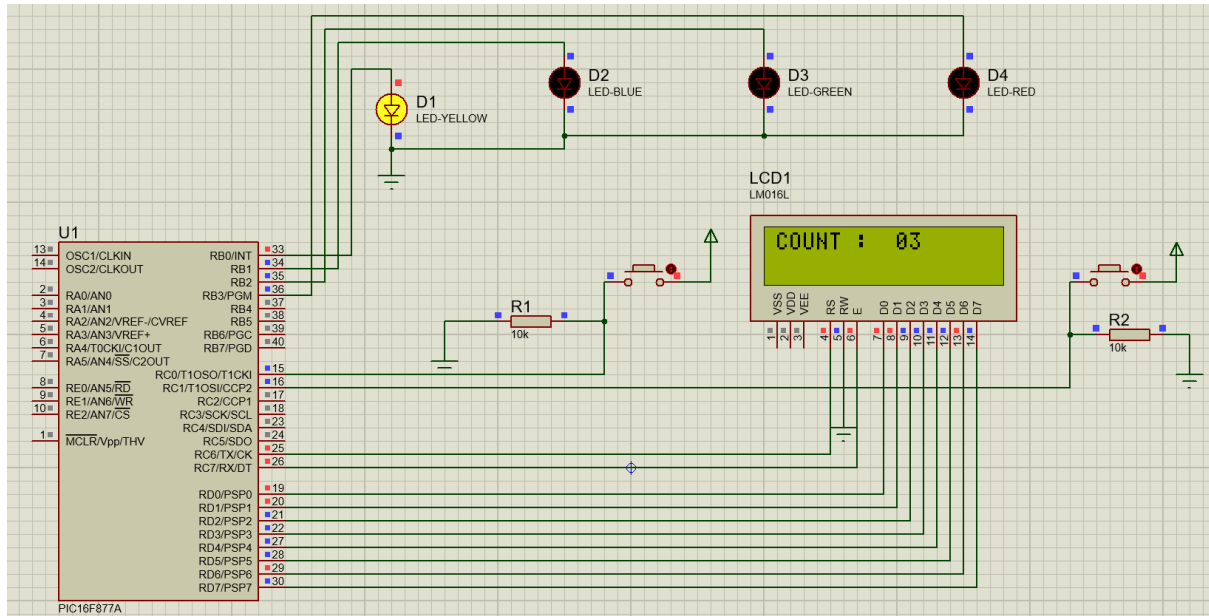
```
{  
LED1=1;  
}  
else  
{  
LED1=0;  
}  
if(count>=5)  
{  
LED2=1;  
}  
else  
{  
LED2=0;  
}  
if(count>=8)  
{  
LED3=1;  
}  
else  
{  
LED3=0;  
}  
if(count>9)  
{  
LED4=1;  
}  
else  
{  
LED4=0;  
}
```

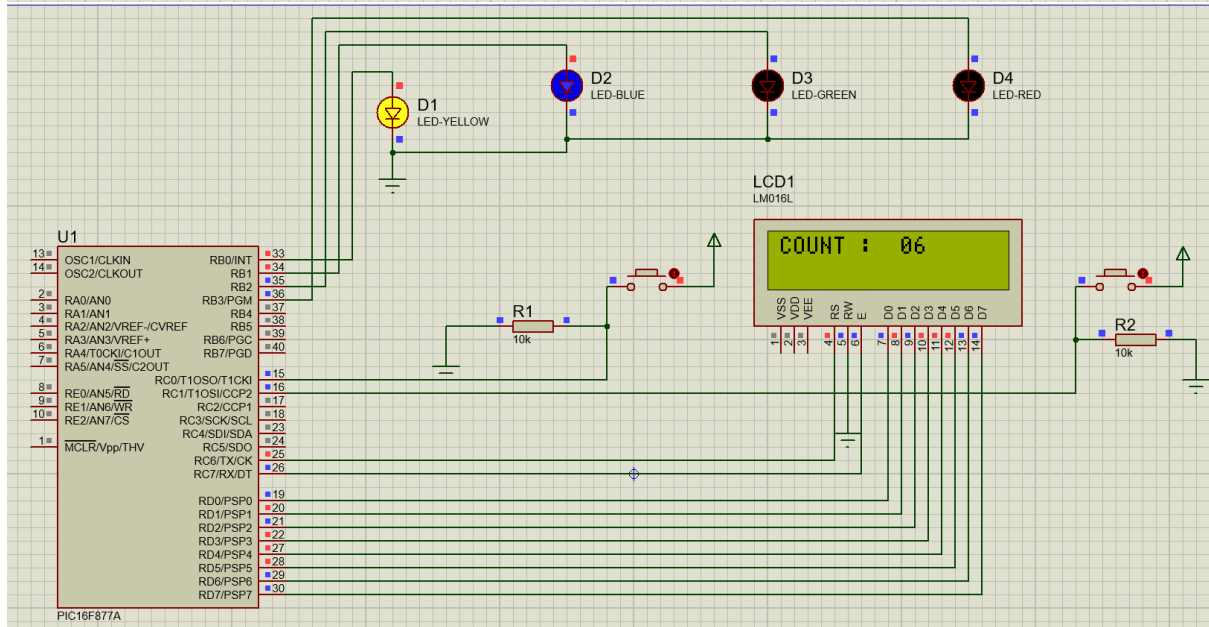
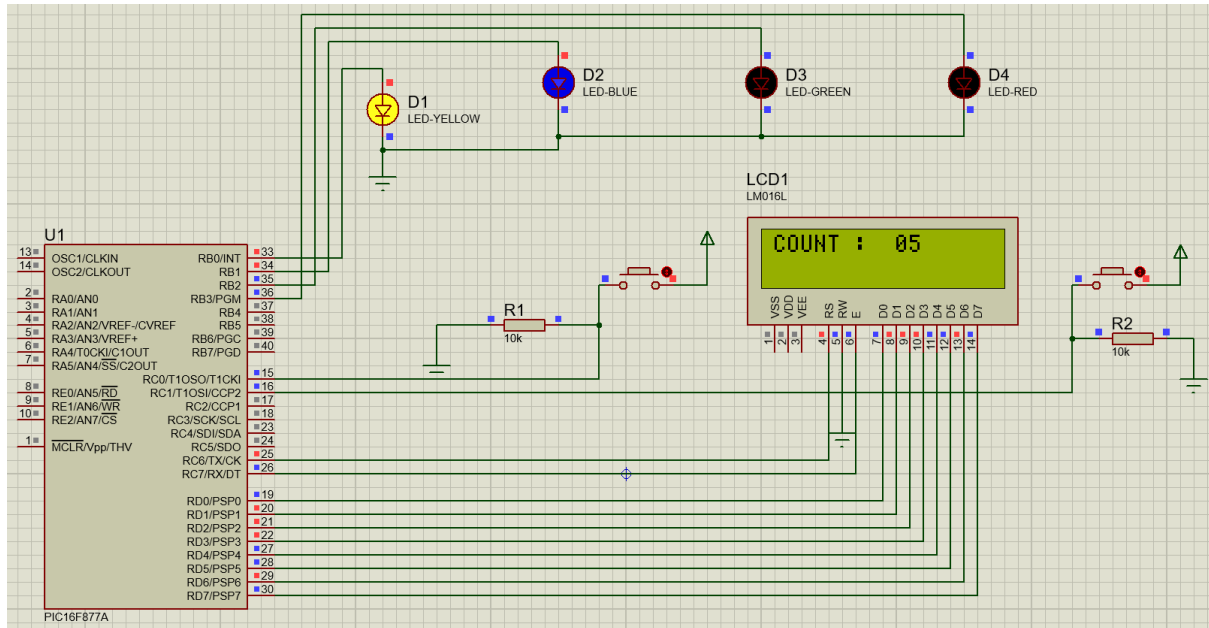
```
}
}
```

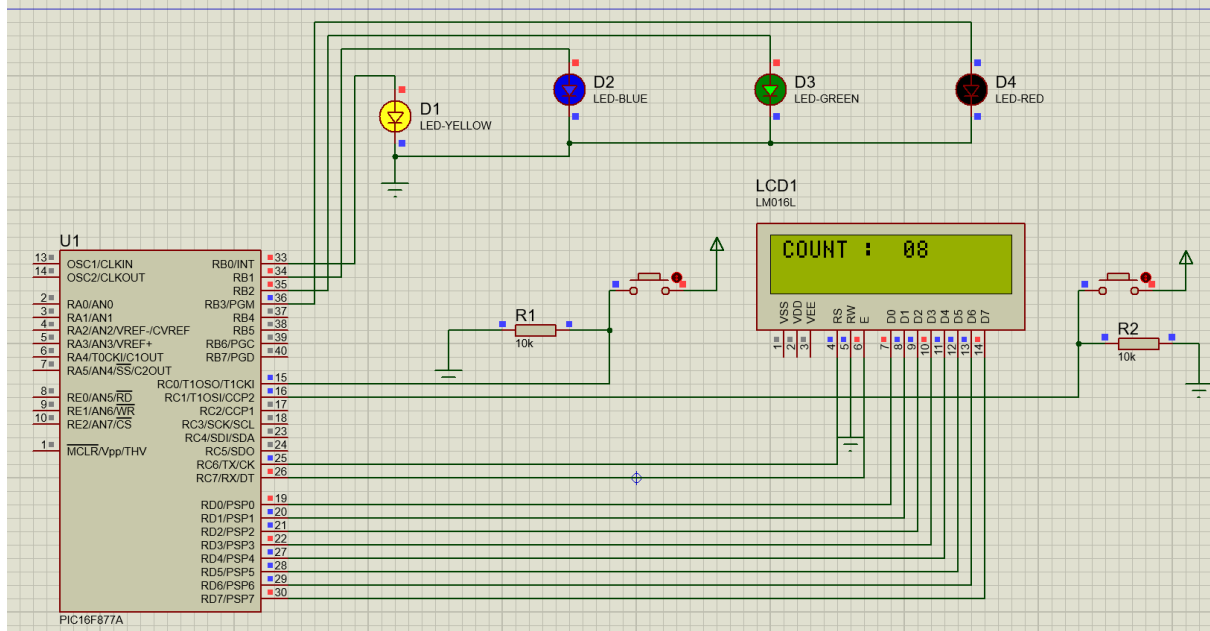
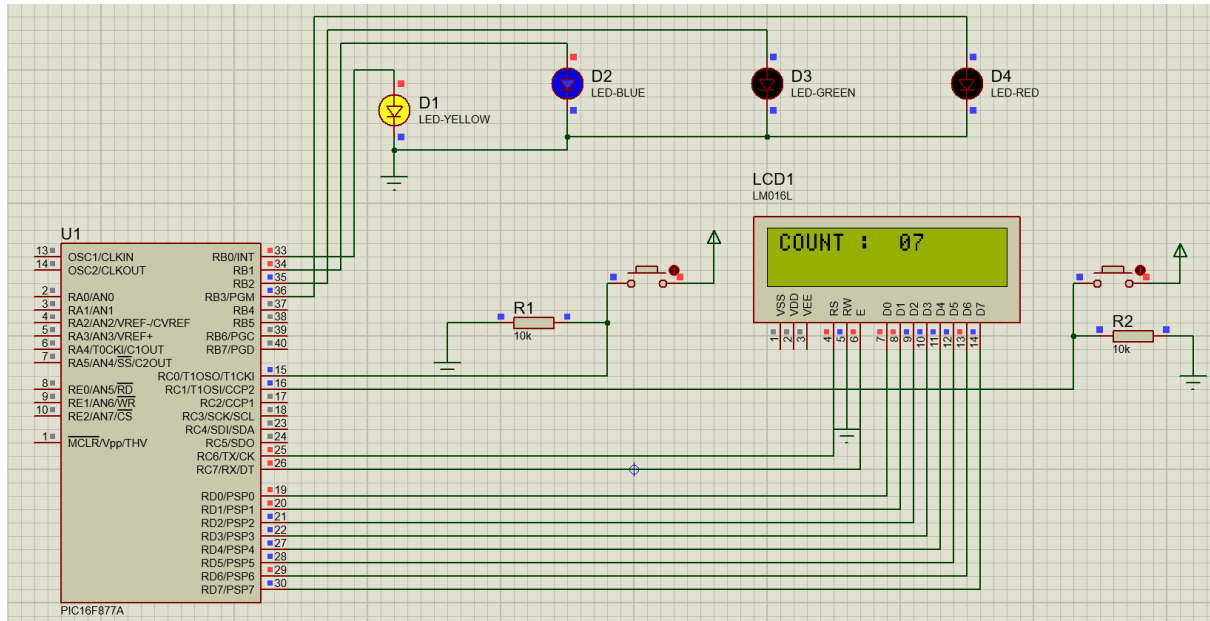
# **SIMULATED OUTPUT:**

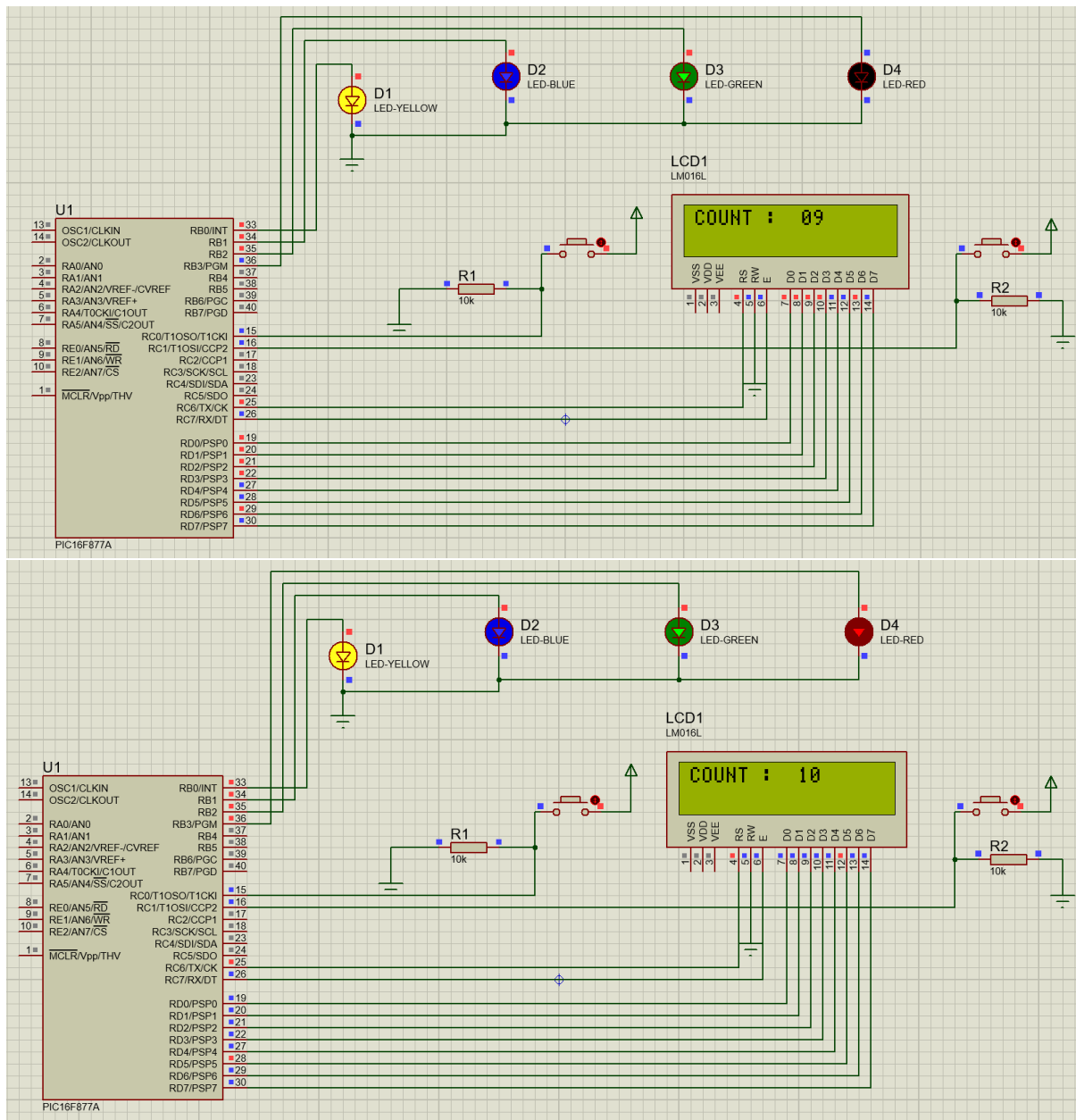












11). Write an embedded C Program when switch-1 is pressed LEDs will be turned On ,when switch-2 is pressed LEDs will be turned OFF,Use 8 LEDs . [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <2 SWITCH 8 LED INC DEC 3.h>
```

```
#bit LED1=0x06.0
```

```
#bit TLED1=0x86.0
```

```
#bit LED2=0x06.1
```

```
#bit TLED2=0x86.1
```

```
#bit LED3=0x06.2
#bit TLED3=0x86.2
#bit LED4=0x06.3
#bit TLED4=0x86.3
#bit LED5=0x06.4
#bit TLED5=0x86.4
#bit LED6=0x06.5
#bit TLED6=0x86.5
#bit LED7=0x06.6
#bit TLED7=0x86.6
#bit LED8=0x06.7
#bit TLED8=0x86.7
#bit SW1=0x07.6
#bit TSW1=0x87.6
#bit SW2=0x07.7
#bit TSW2=0x87.7
```

```
int count=0;
```

```
void main()
```

```
{
```

```
LED1=0;
```

```
TLED1=0;
```

```
LED2=0;
```

```
TLED2=0;
```

```
LED3=0;
```

```
TLED3=0;
```

```
LED4=0;
```

```
TLED4=0;
```

```
LED5=0;
```

```
TLED5=0;
```

```
LED6=0;
```



TLED6=0;

LED7=0;

TLED7=0;

LED8=0;

TLED8=0;

SW1=0;

TSW1=1;

SW2=0;

TSW2=1;

while(TRUE)

{

if(SW1==1)

{

while(SW1==1);

count+=1;

}

if(SW2==1)

{

while(SW2==1);

count-=1;

}

if(count>=1 && count<=8)

{

LED1=1;

delay\_ms(200);

}

else

{

LED1=0;

```
delay_ms(200);  
}  
if(count>=2 && count<=8)  
{  
    LED2=1;  
    delay_ms(200);  
}  
else  
{  
    LED2=0;  
    delay_ms(200);  
}  
if(count>=3 && count<=8)  
{  
    LED3=1;  
    delay_ms(200);  
}  
else  
{  
    LED3=0;  
    delay_ms(200);  
}  
if(count>=4 && count<=8)  
{  
    LED4=1;  
    delay_ms(200);  
}  
else  
{  
    LED4=0;  
    delay_ms(200);
```

```
}  
  
if(count>=5 && count<=8)  
{  
    LED5=1;  
    delay_ms(200);  
}  
  
else  
{  
    LED5=0;  
    delay_ms(200);  
}  
  
if(count>=6 && count<=8)  
{  
    LED6=1;  
    delay_ms(200);  
}  
  
else  
{  
    LED6=0;  
    delay_ms(200);  
}  
  
if(count>=7 && count<=8)  
{  
    LED7=1;  
    delay_ms(200);  
}  
  
else  
{  
    LED7=0;  
    delay_ms(200);  
}
```

```

if(count==8)
{
    LED8=1;

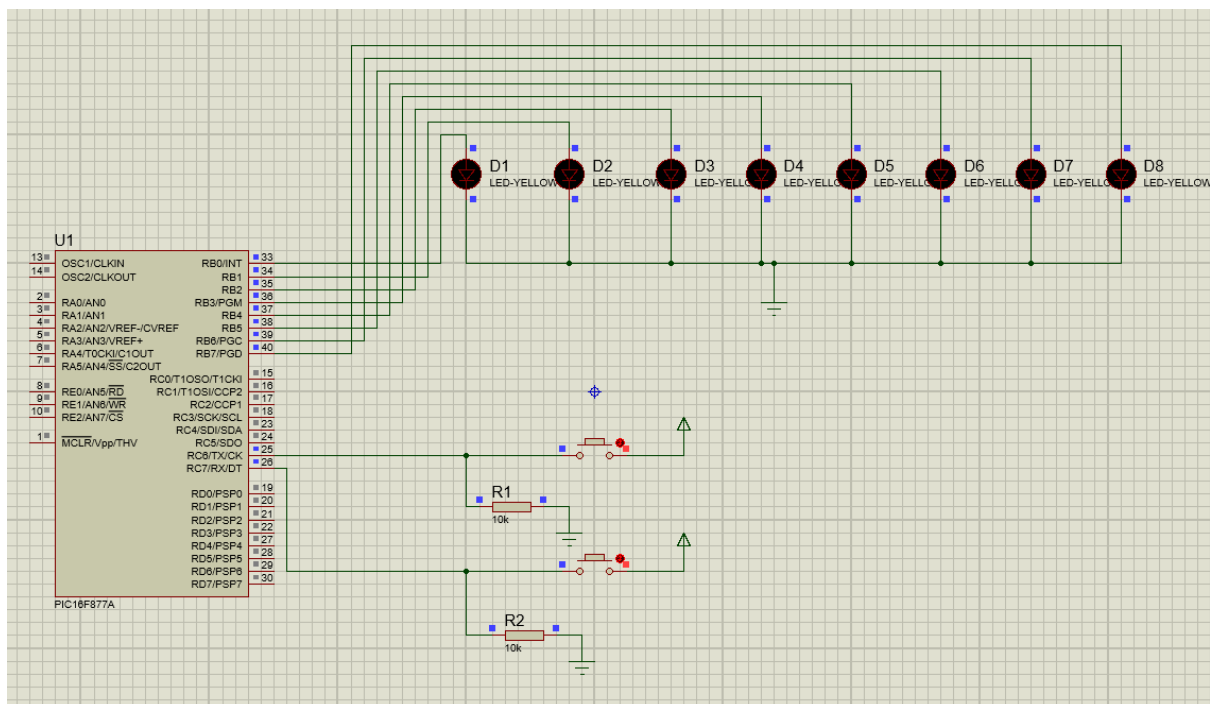
    delay_ms(200);
}

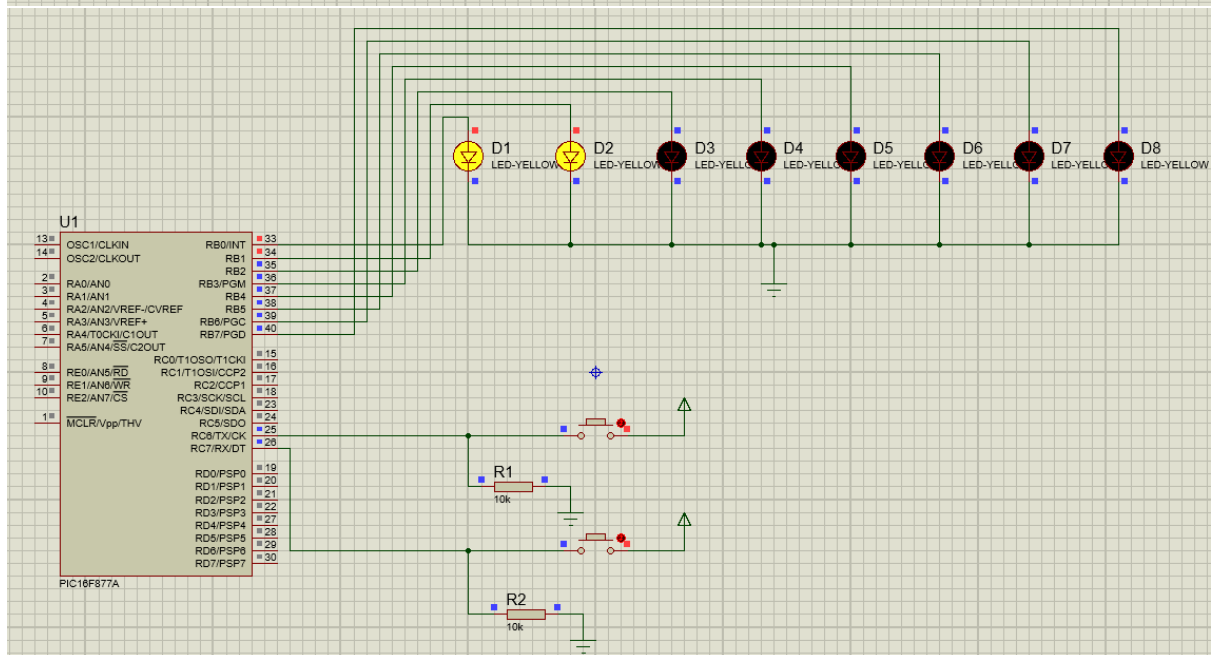
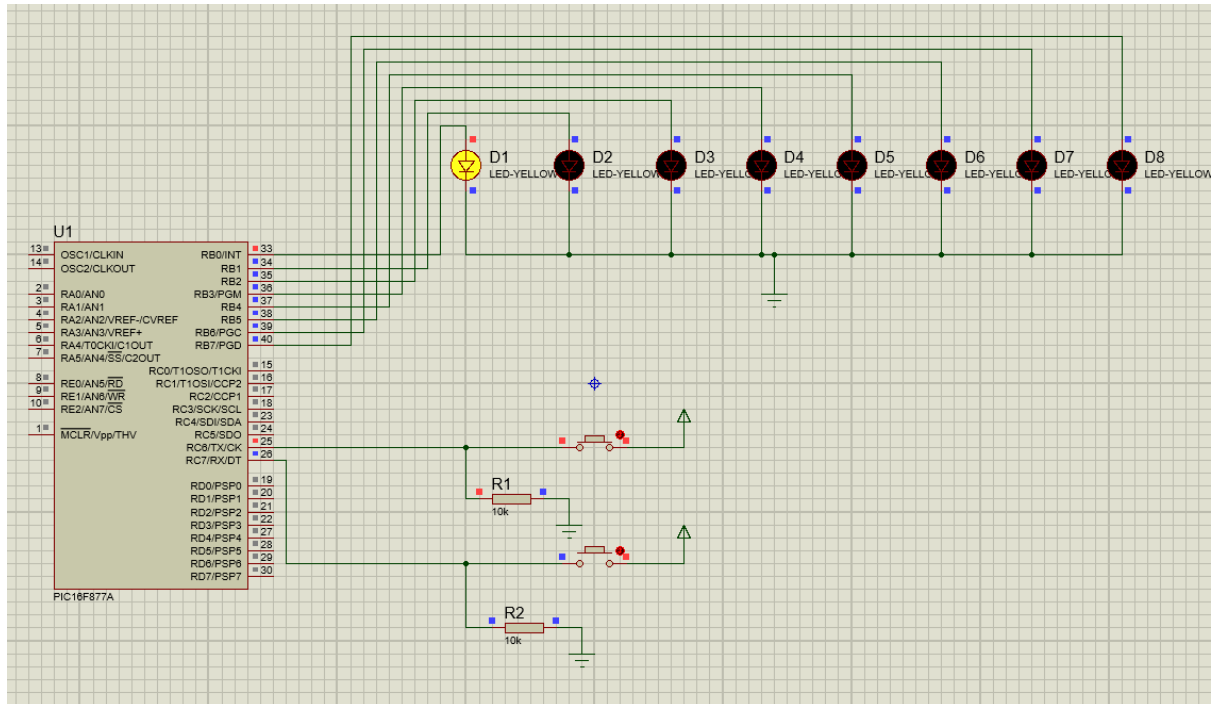
else
{
    LED8=0;

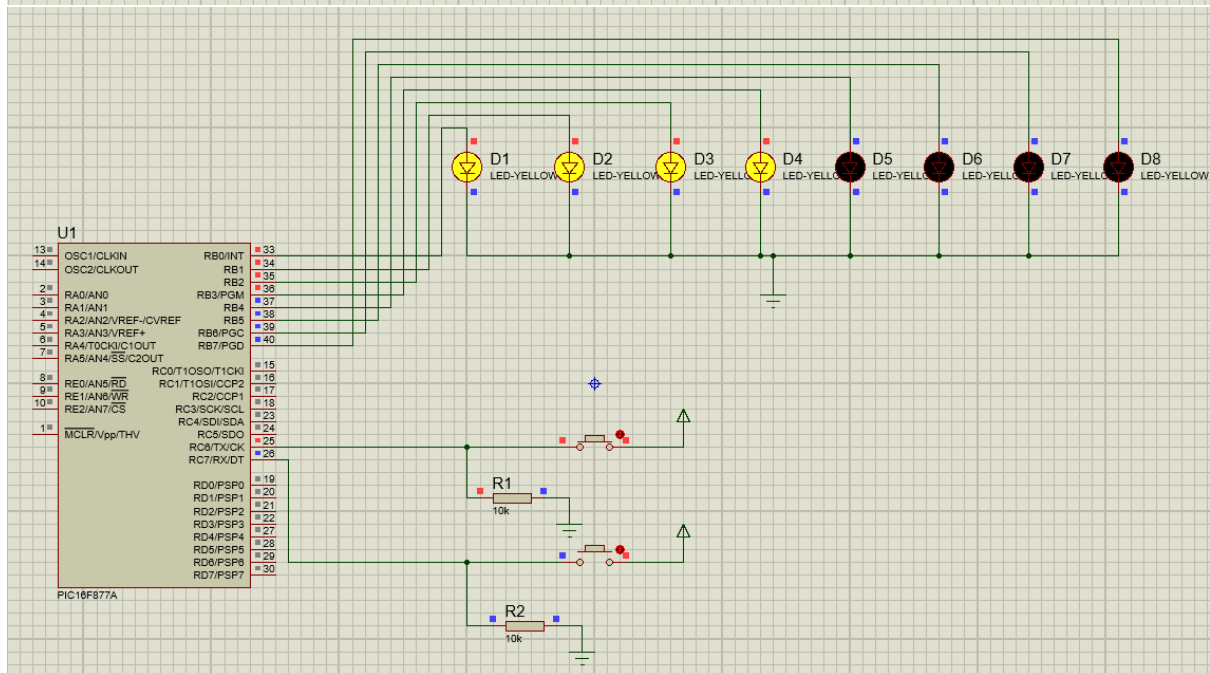
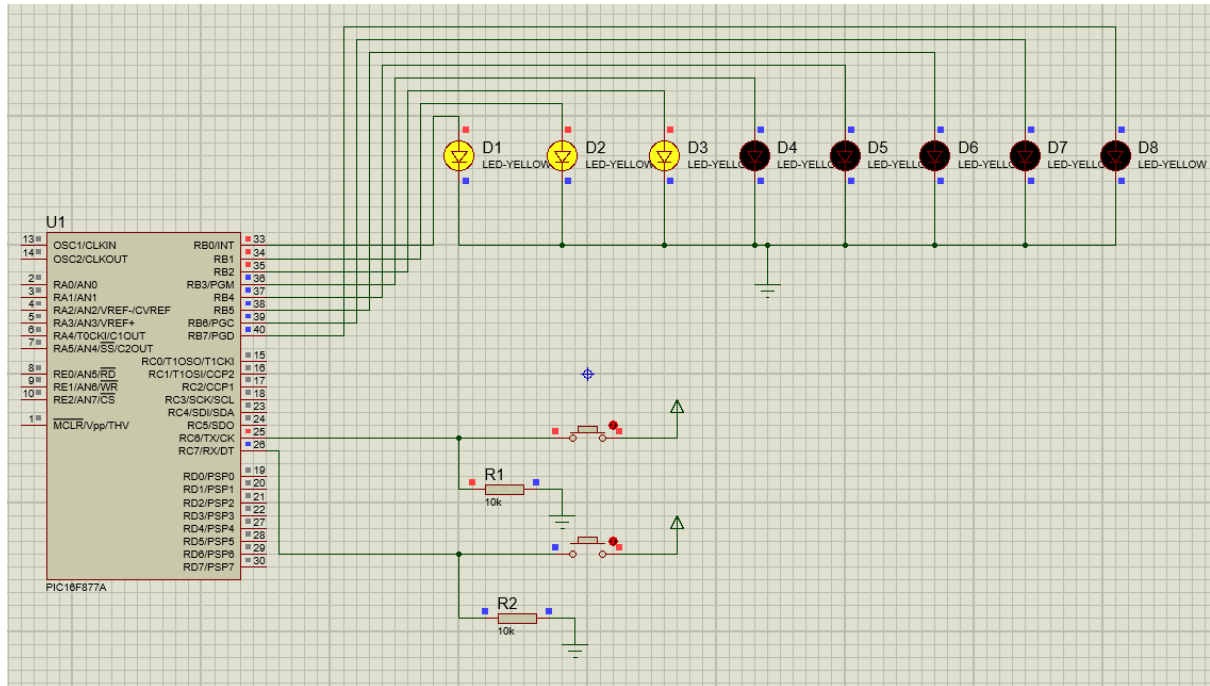
    delay_ms(200);
}
}
}

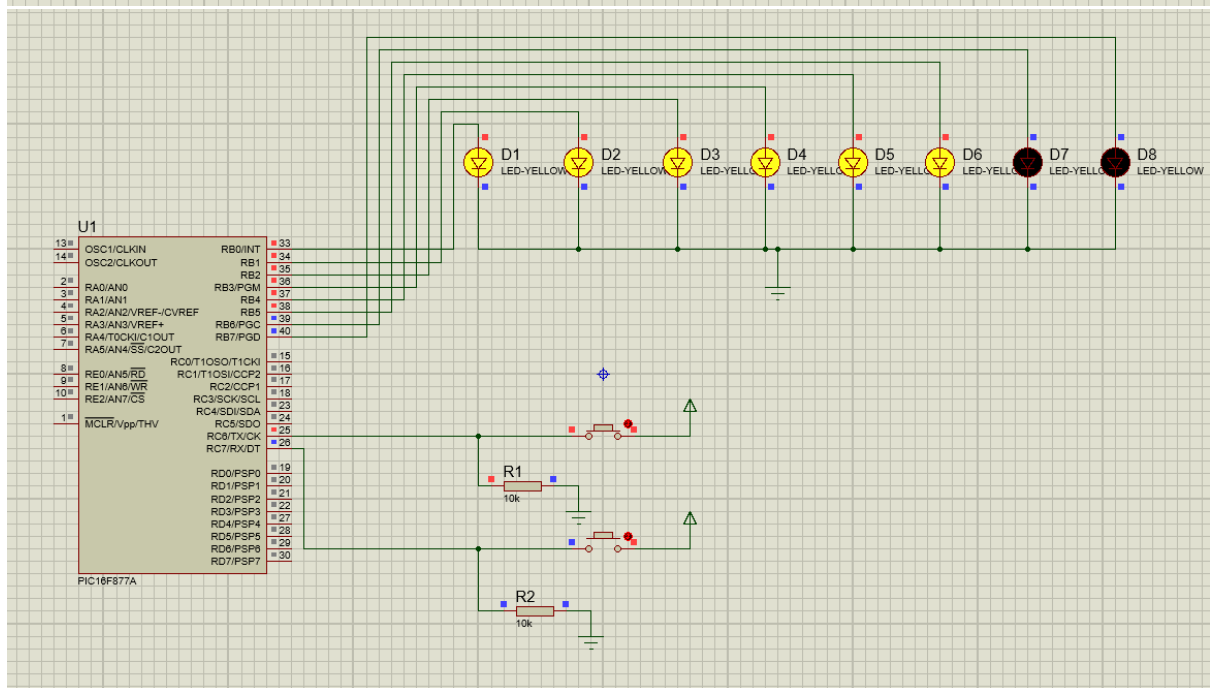
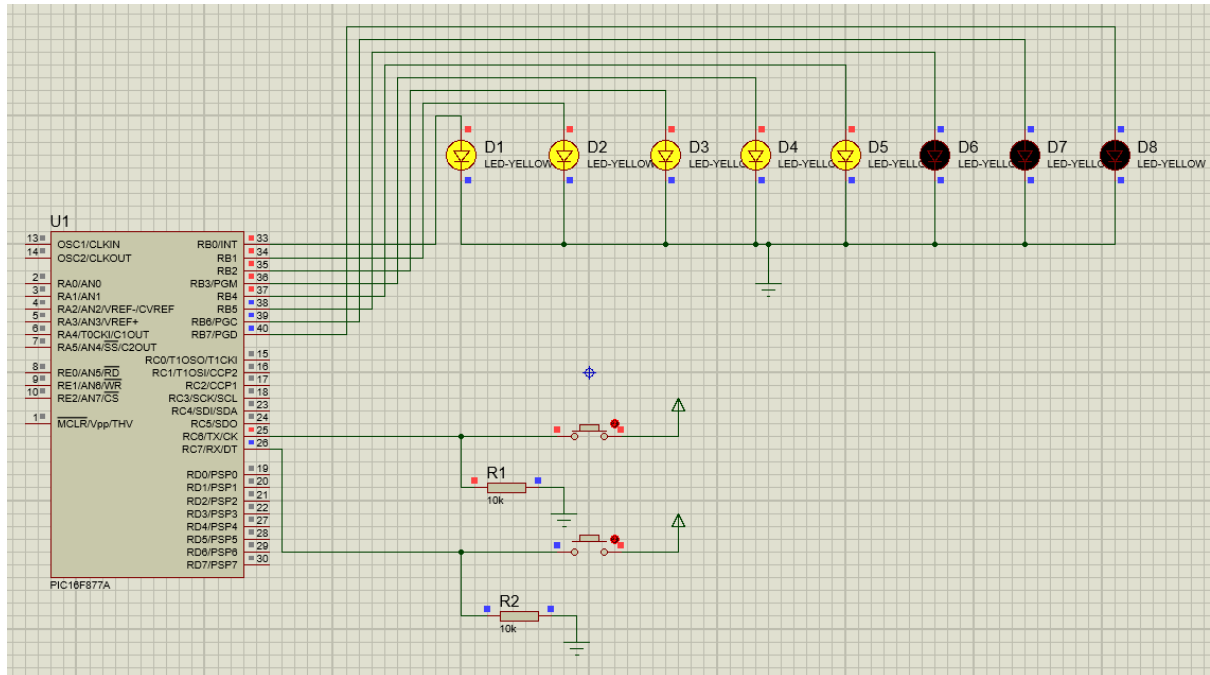
```

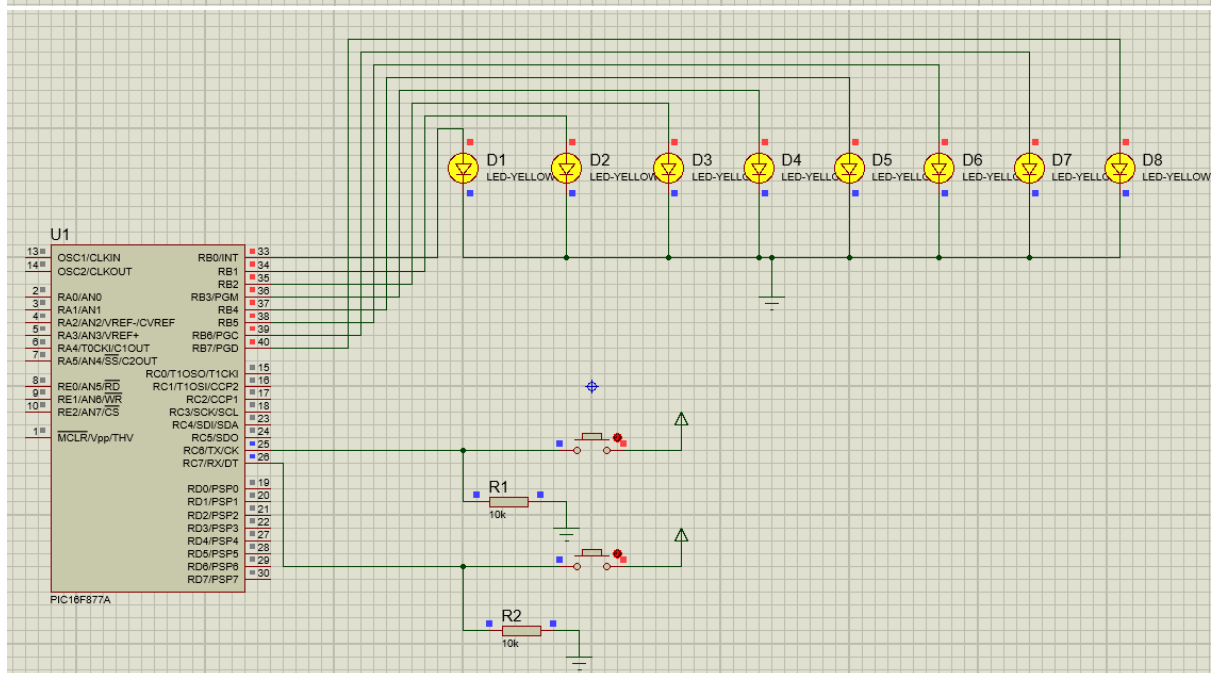
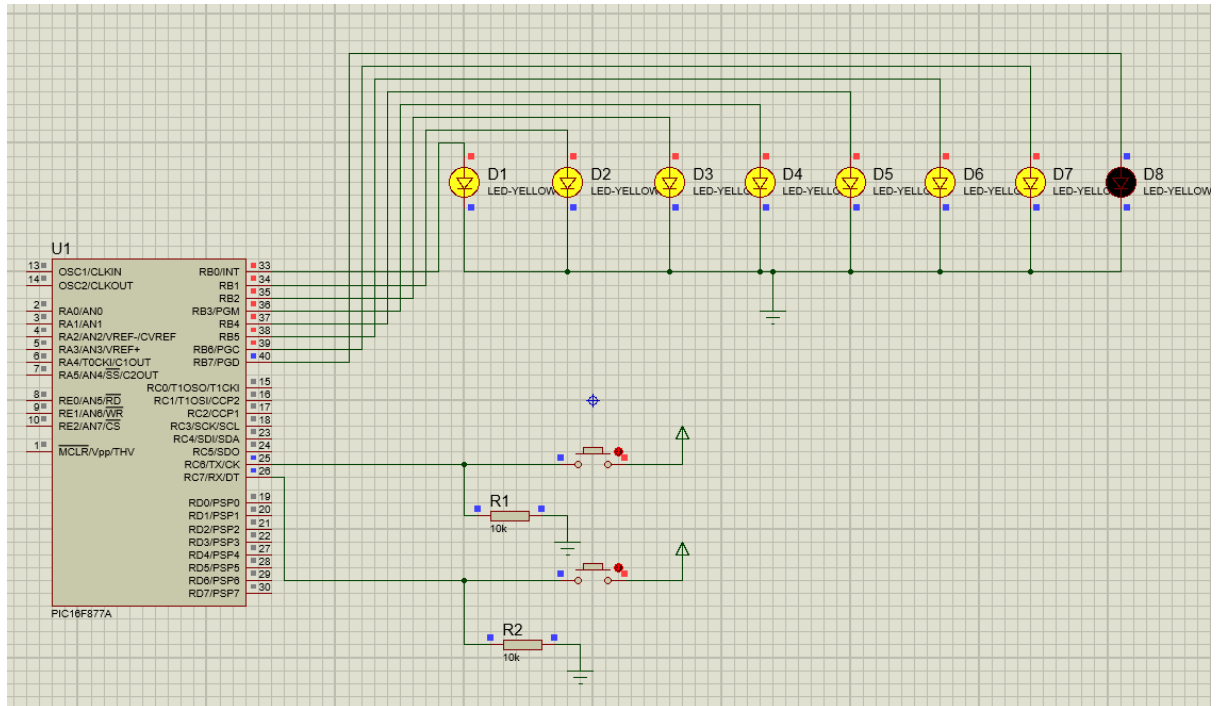
### SIMULATED OUTPUT:



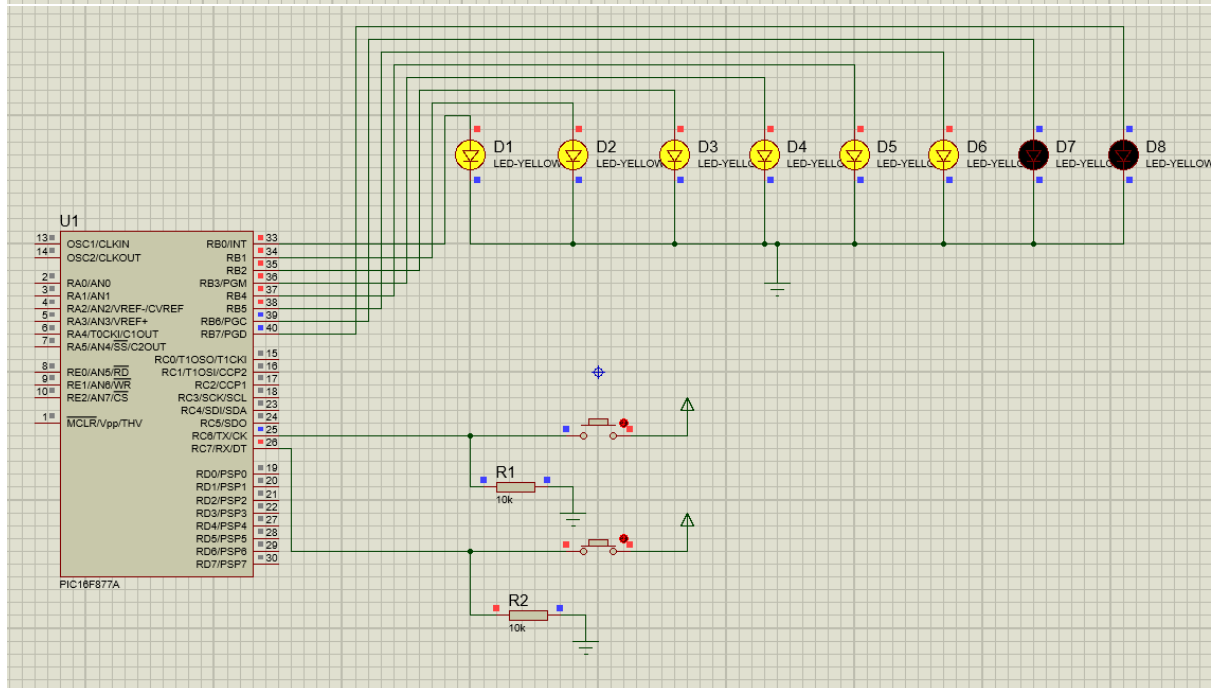
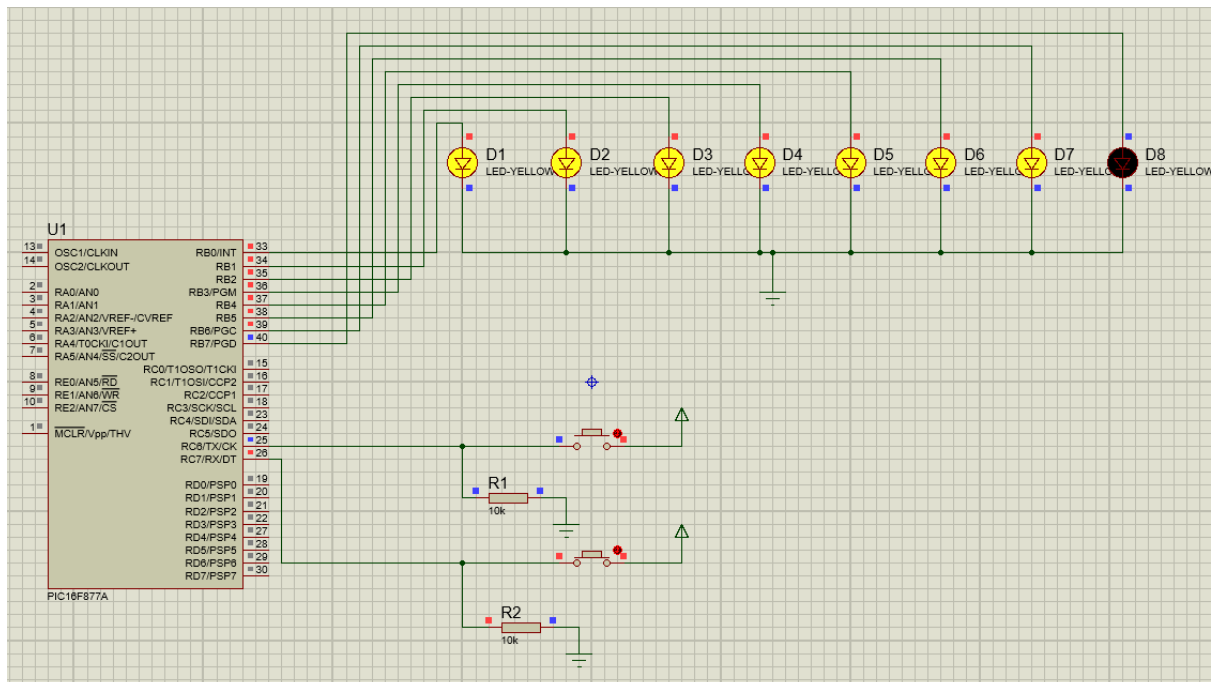


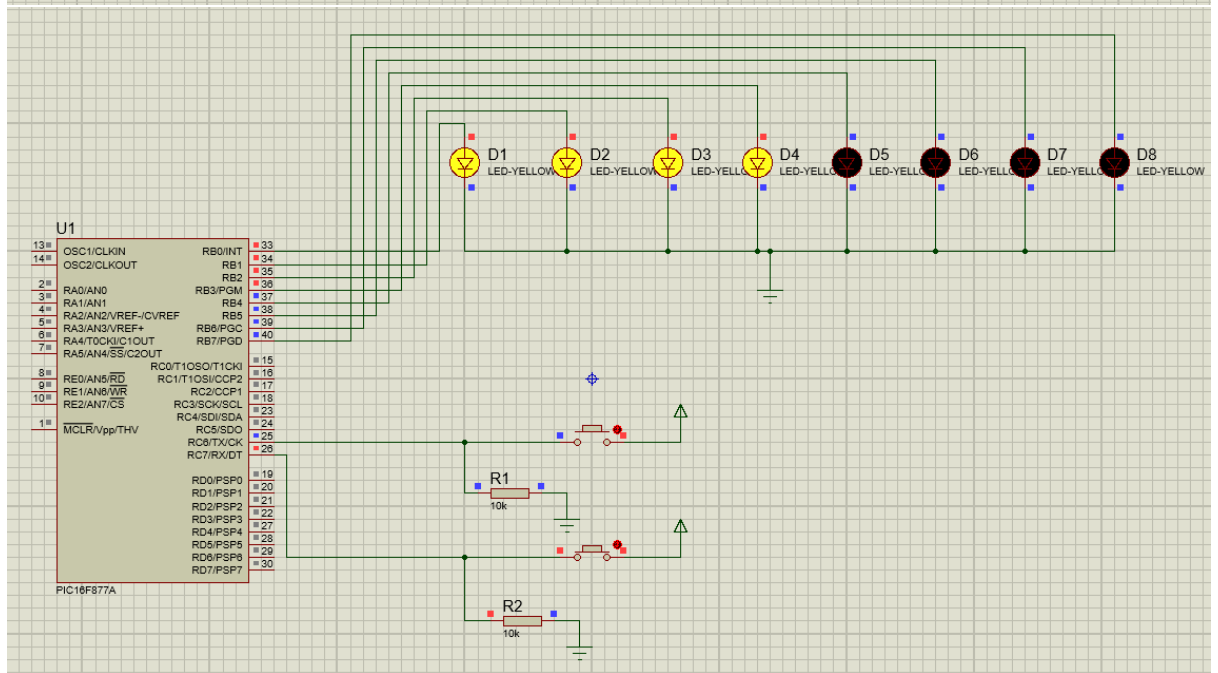
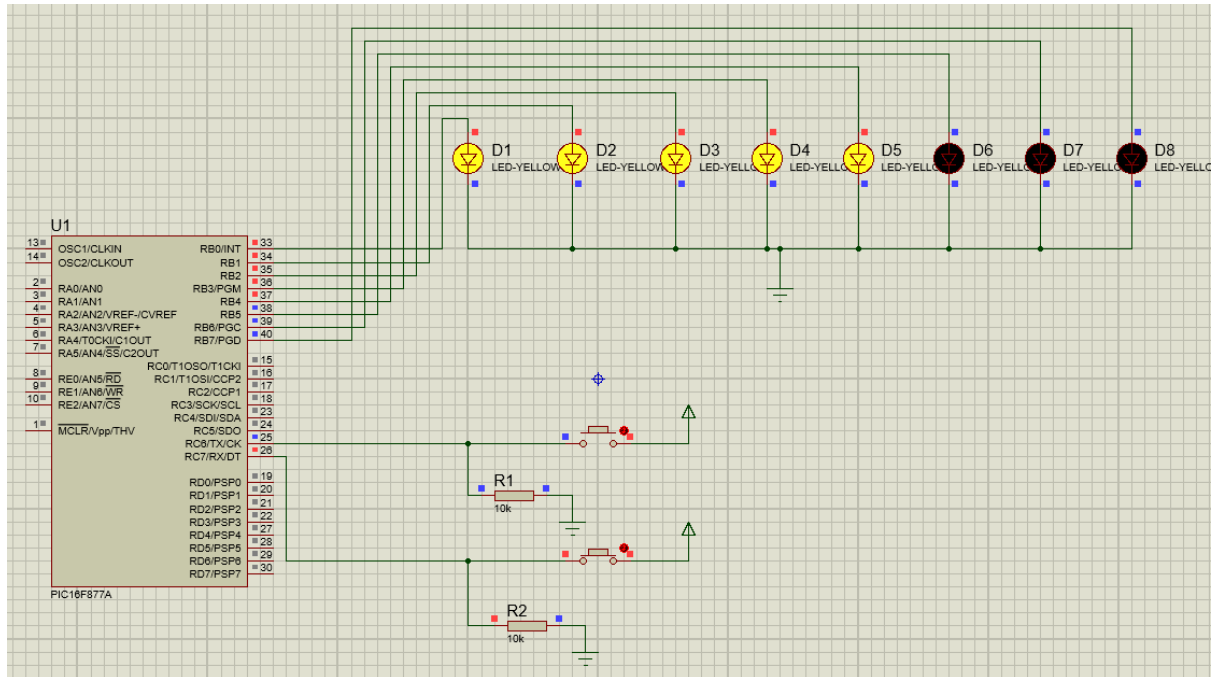


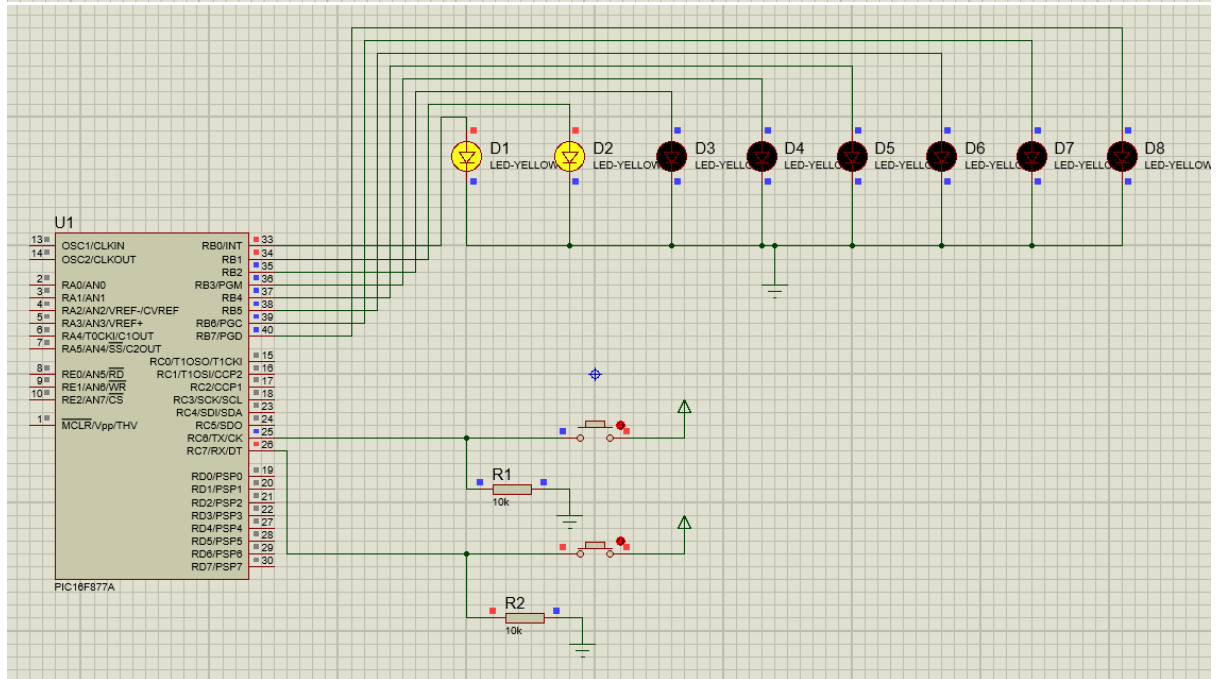
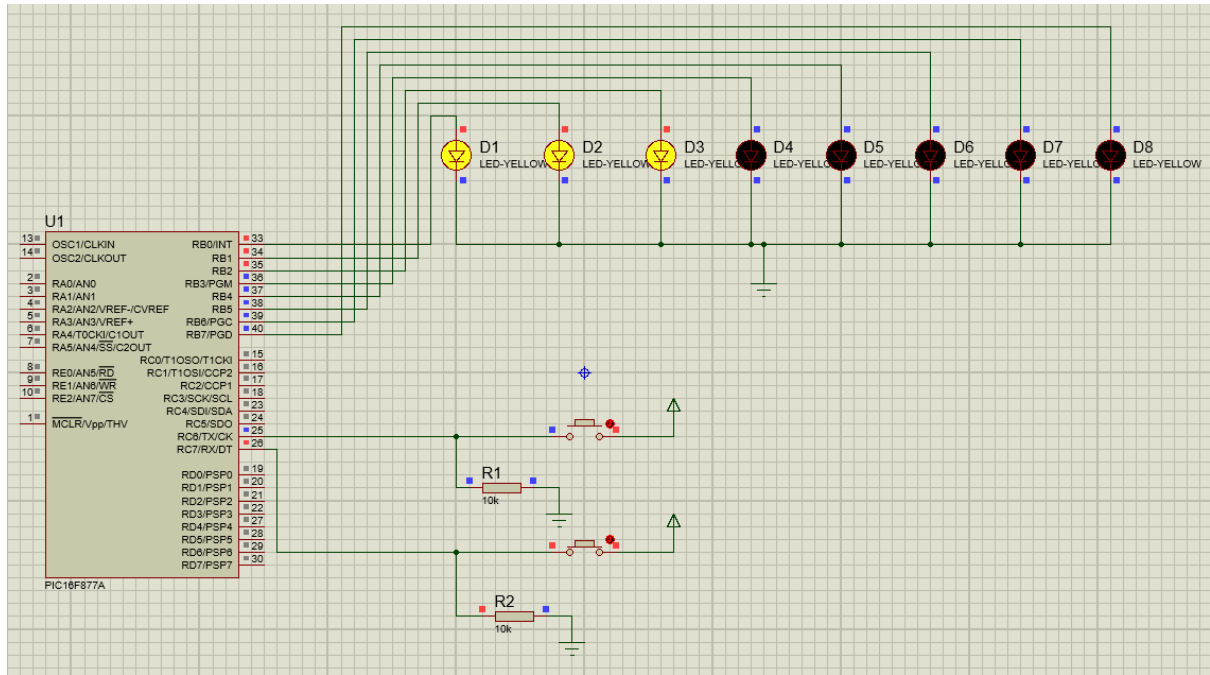


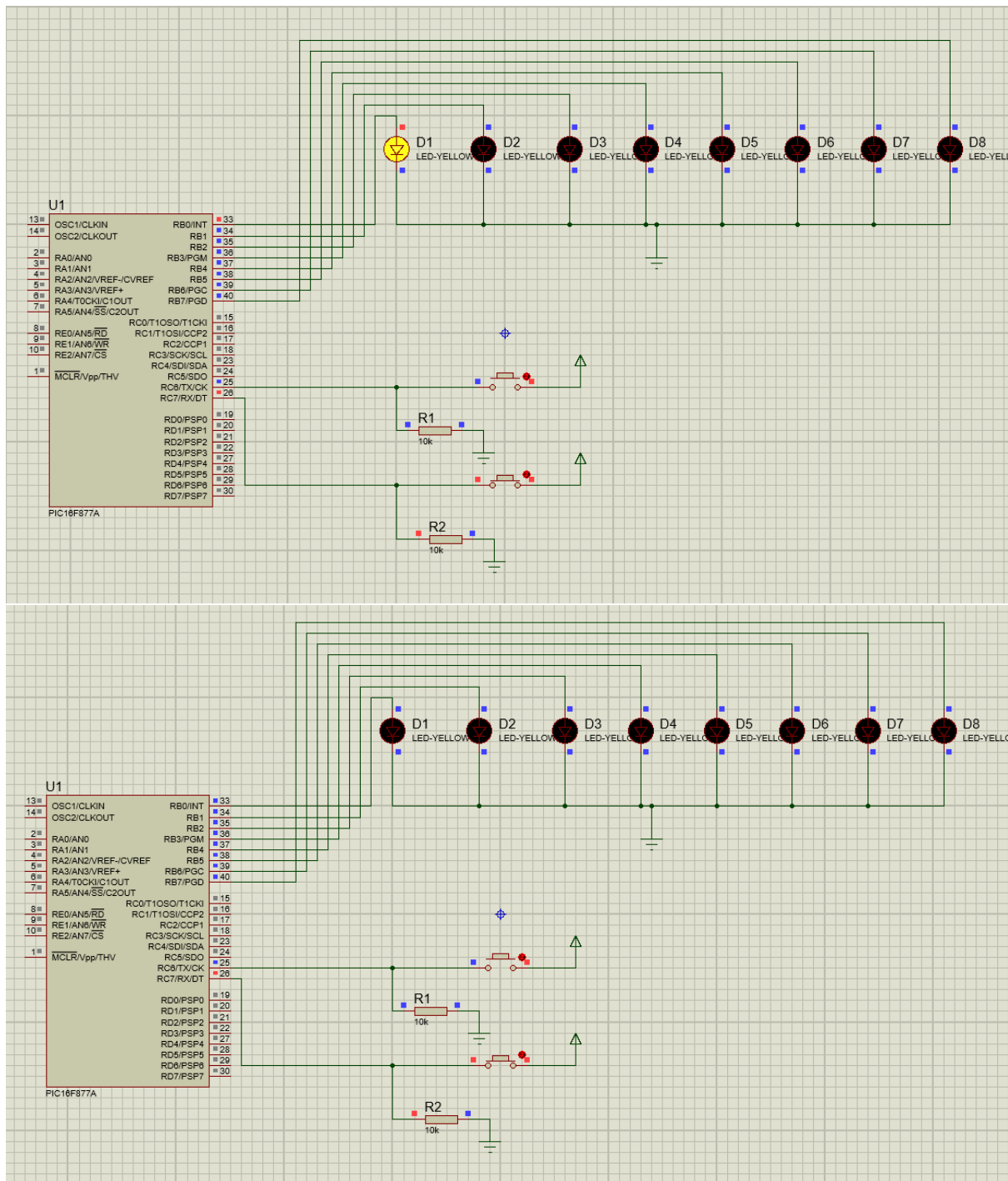












12). Write an embedded C Program for ADC . [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <ADC1.h>
```

```
#byte LCD=0x08
```

#byte TLCD=0x88

#bit RS=0x07.6

#bit TRS=0x87.6

#bit EN=0x07.7

#bit TEN=0x87.7

#byte ADCON0=0x1F

#byte ADRESH=0x1E

#byte ADCON1=0x9F

#byte ADRESL=0x9E

#bit GO\_DOWN=0x1F.2

void display(char,int);

void main()

{

ADCON0=0x01;

LCD=0x00;

RS=0;

EN=0;

TEN=0;

TLCD=0;

TRS=0;

ADCON1=0x80;

long int RESULT,adres;

int a,b,c,d,e,f;

```

while(TRUE)
{
    GO_DOWN=1;
    display(0x38,0);
    display(0x0e,0);
    display(0xc0,0);
    adres=ADRESH;
    delay_ms(200);
    RESULT=ADRESL|adres<<8;
    delay_ms(200);

    a=RESULT/1000;
    b=RESULT%1000;
    c=b/100;
    d=b%100;
    e=d/10;
    f=d%10;
    display(a+0x30,1);
    delay_ms(200);
    display(c+0x30,1);
    delay_ms(200);
    display(e+0x30,1);
    delay_ms(200);
    display(f+0x30,1);
    delay_ms(200);
}

}

```

```

void display(char x,int y)
{

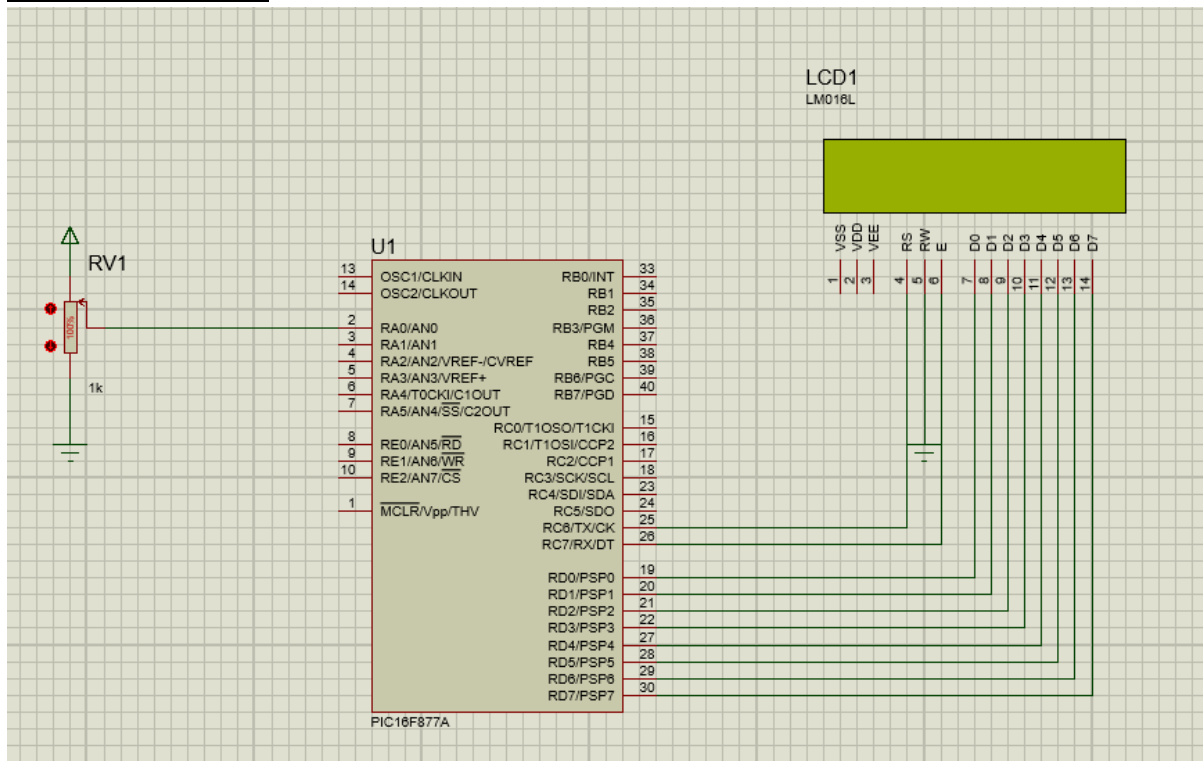
```

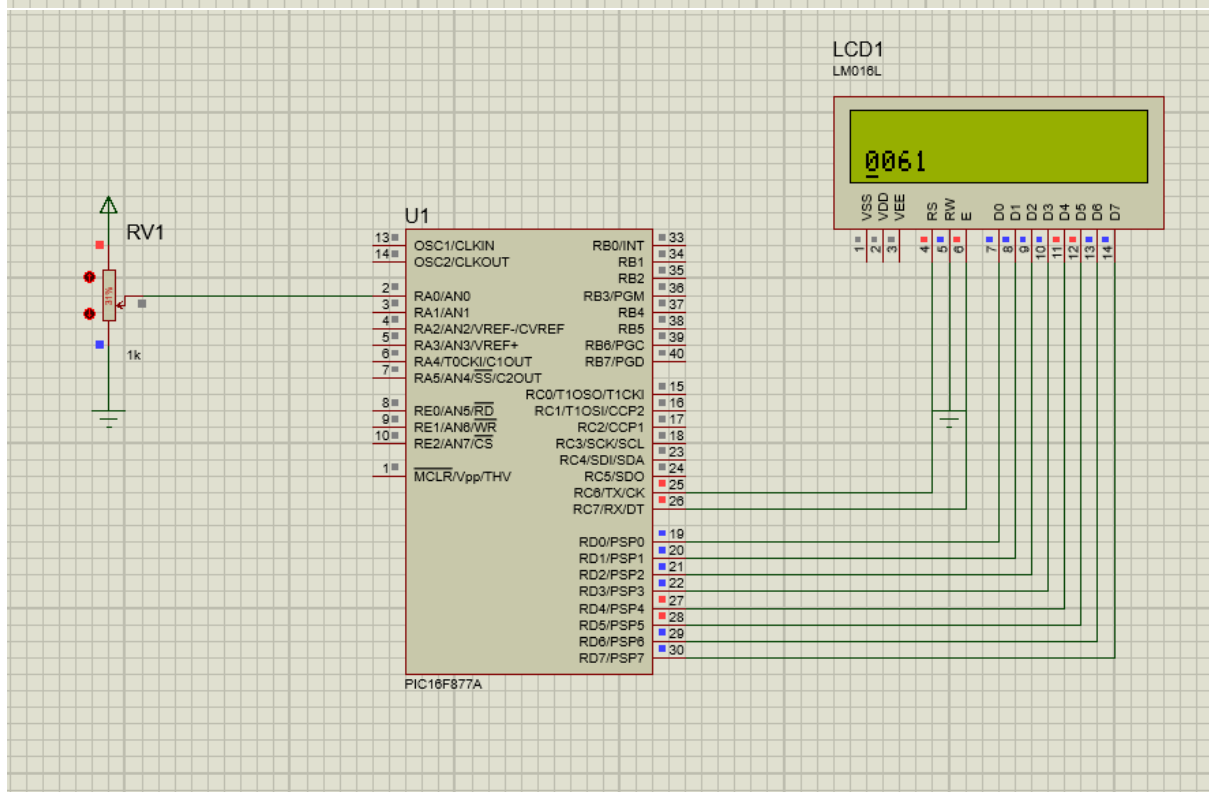
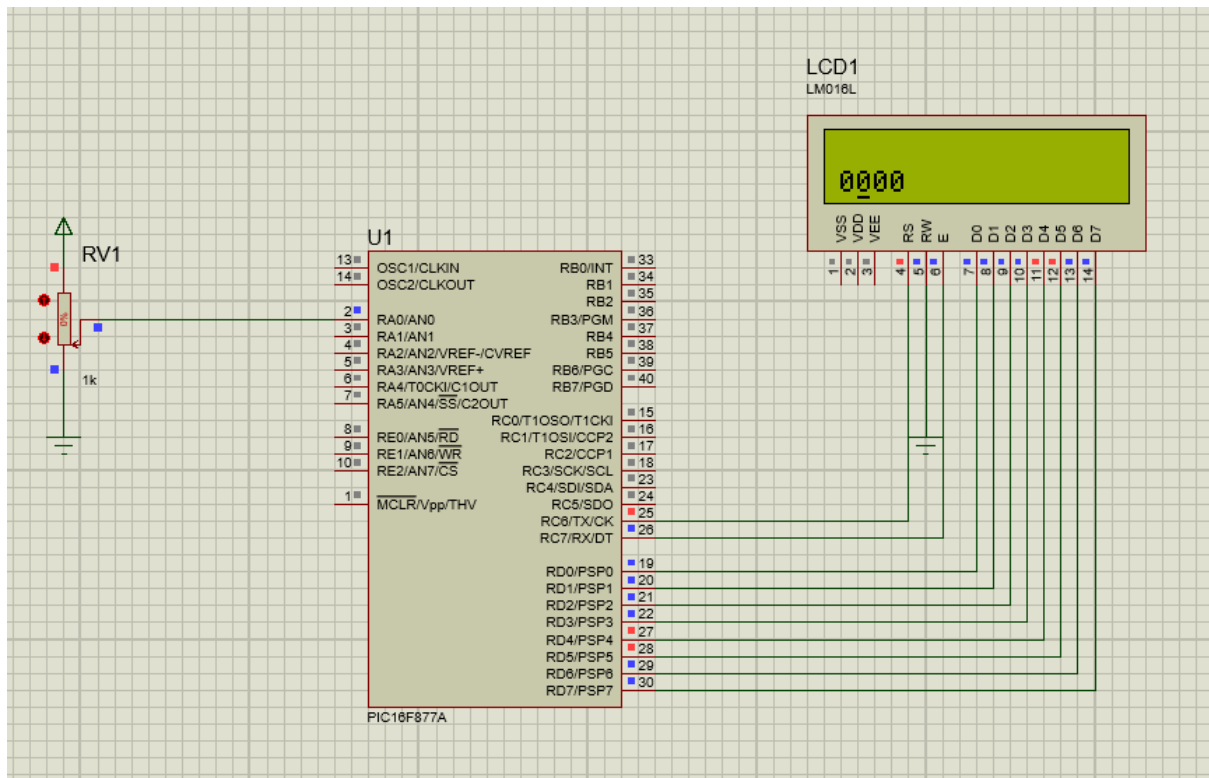
```

LCD=x;
RS=y;
EN=1;
delay_ms(3);
EN=0;
delay_ms(3);
}

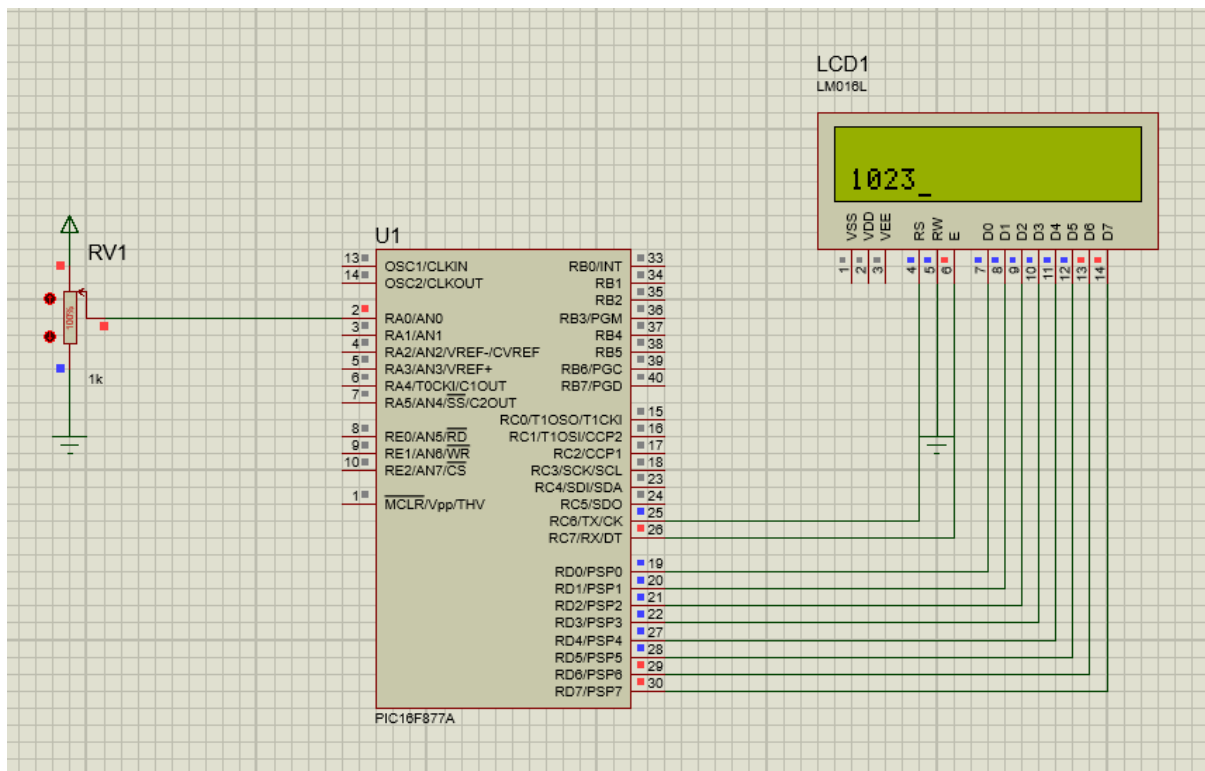
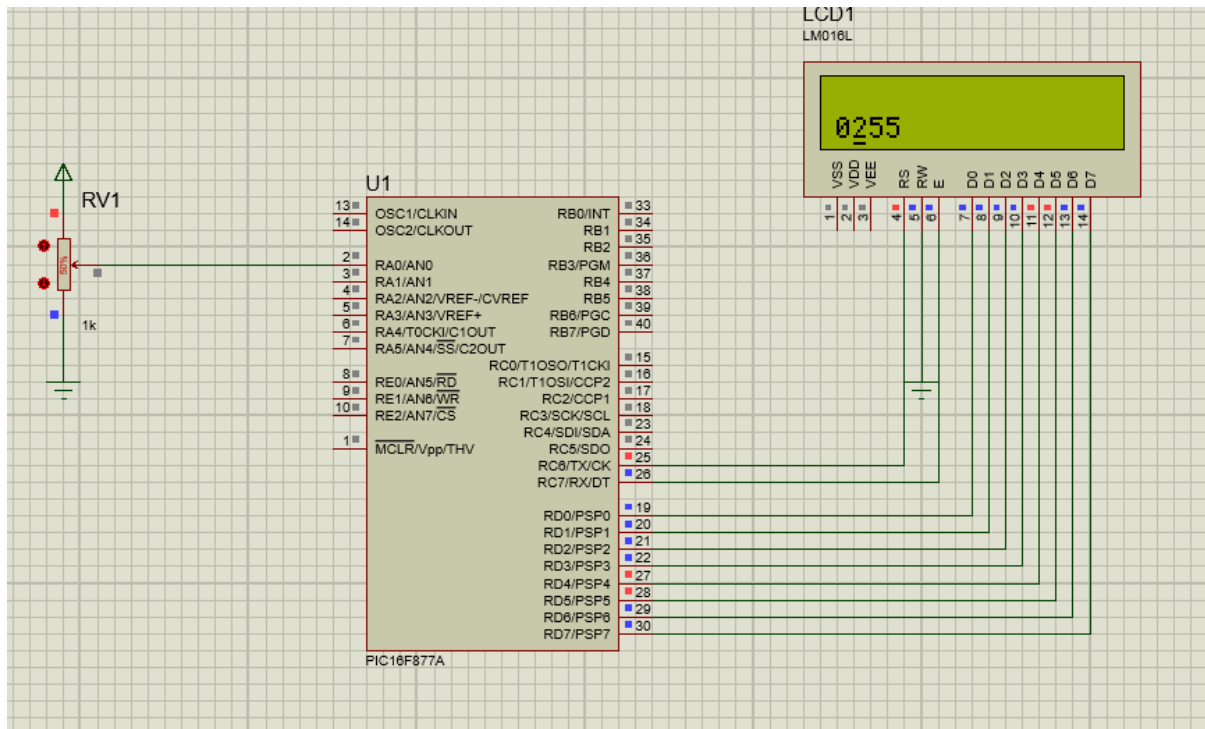
```

### SIMULATED OUTPUT:









13). Write an embedded C Program for turning ON the LEDs, when ADC value greater than zero LED-1 will be turned On, when ADC value greater than 250 LED-2 will be turned On, when ADC value greater than 500 LED-3 will be turned On, when ADC value greater than 1000 LED-4 will be turned On. [ complete the simulation?] / execute this using proteus 8 Professional Software?

A).

```
#include <4 LED ADC count.h>
```

```
#byte LCD=0x08
```

```
#byte TLCD=0x88
```

```
#bit RS=0x07.6
```

```
#bit TRS=0x87.6
```

```
#bit EN=0x07.7
```

```
#bit TEN=0x87.7
```

```
#bit LED1=0x06.0
```

```
#bit TLED1=0x86.0
```

```
#bit LED2=0x06.1
```

```
#bit TLED2=0x86.1
```

```
#bit LED3=0x06.2
```

```
#bit TLED3=0x86.2
```

```
#bit LED4=0x06.3
```

```
#bit TLED4=0x86.3
```

```
#byte ADCON0=0x1F
```

```
#byte ADRESH=0x1E
```

```
#byte ADCON1=0x9F
```

```
#byte ADRESL=0x9E
```

```
#bit GO_DOWN=0x1F.2
```

```
void display(char ,int);
```

```
void main()
```

```
{
```

```
ADCON0=0x01;
```

```
LCD=0x00;
```

```
RS=0;
```

```
EN=0;
```

```
TEN=0;
```

```
TLCD=0;
```

```
TRS=0;
```

```
LED1=0;
```

```
TLED1=0;
```

```
LED2=0;
```

```
TLED2=0;
```

```
LED3=0;
```

```
TLED3=0;
```

```
LED4=0;
```

```
TLED4=0;
```

```
ADCON1=0x80;
```

```
long int RESULT,adres;
```

```
int a,b,c,d,e,f;
```

```
while(TRUE)
```

```
{
```

```
GO_DOWN=1;
```

```
display(0x38,0);
```

```
display(0x0e,0);
```

```
display(0xc0,0);
```

```
adres=ADRESH;
```

```
delay_ms(200);  
RESULT=ADRESL|adres<<8;  
delay_ms(200);
```

```
a=RESULT/1000;  
b=RESULT%1000;  
c=b/100;  
d=b%100;  
e=d/10;  
f=d%10;  
display(a+0x30,1);  
delay_ms(200);  
display(c+0x30,1);  
delay_ms(200);  
display(e+0x30,1);  
delay_ms(200);  
display(f+0x30,1);  
delay_ms(200);
```

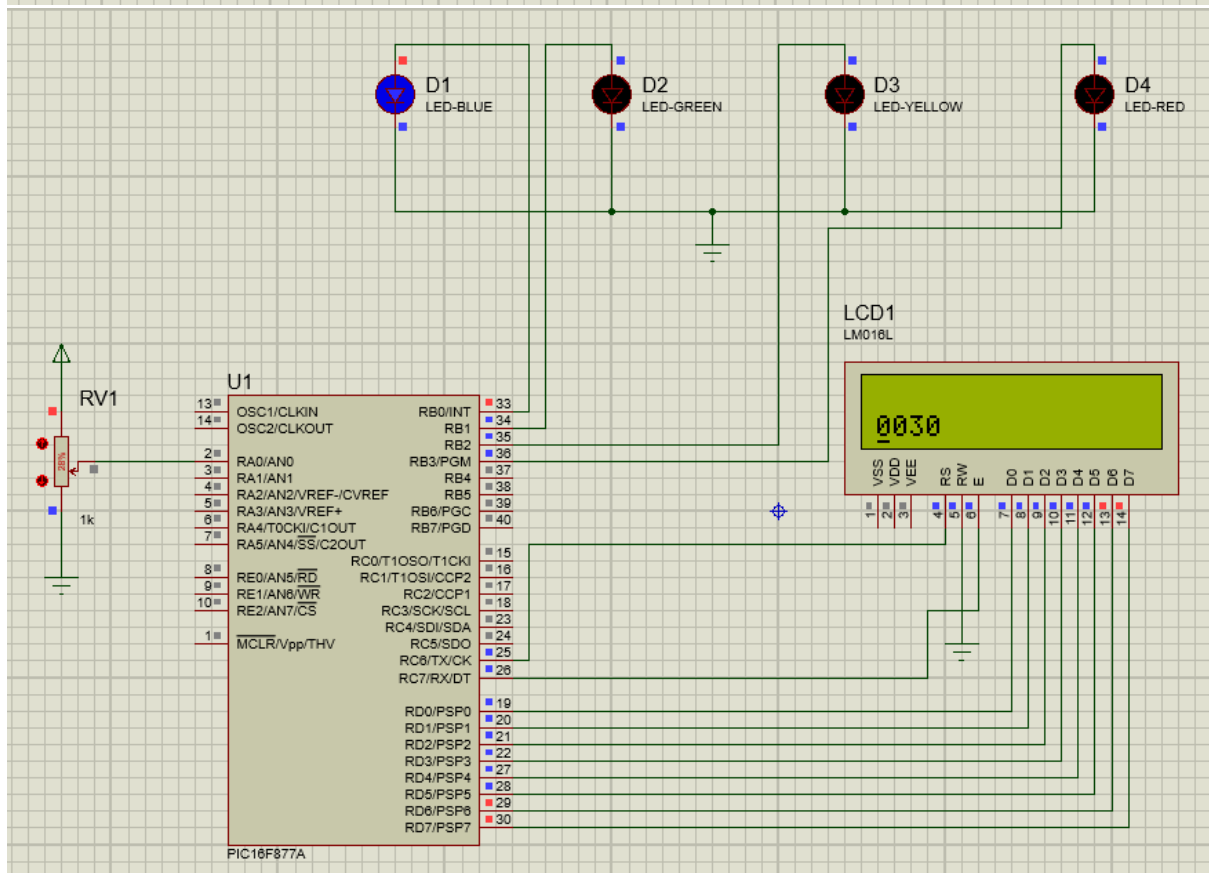
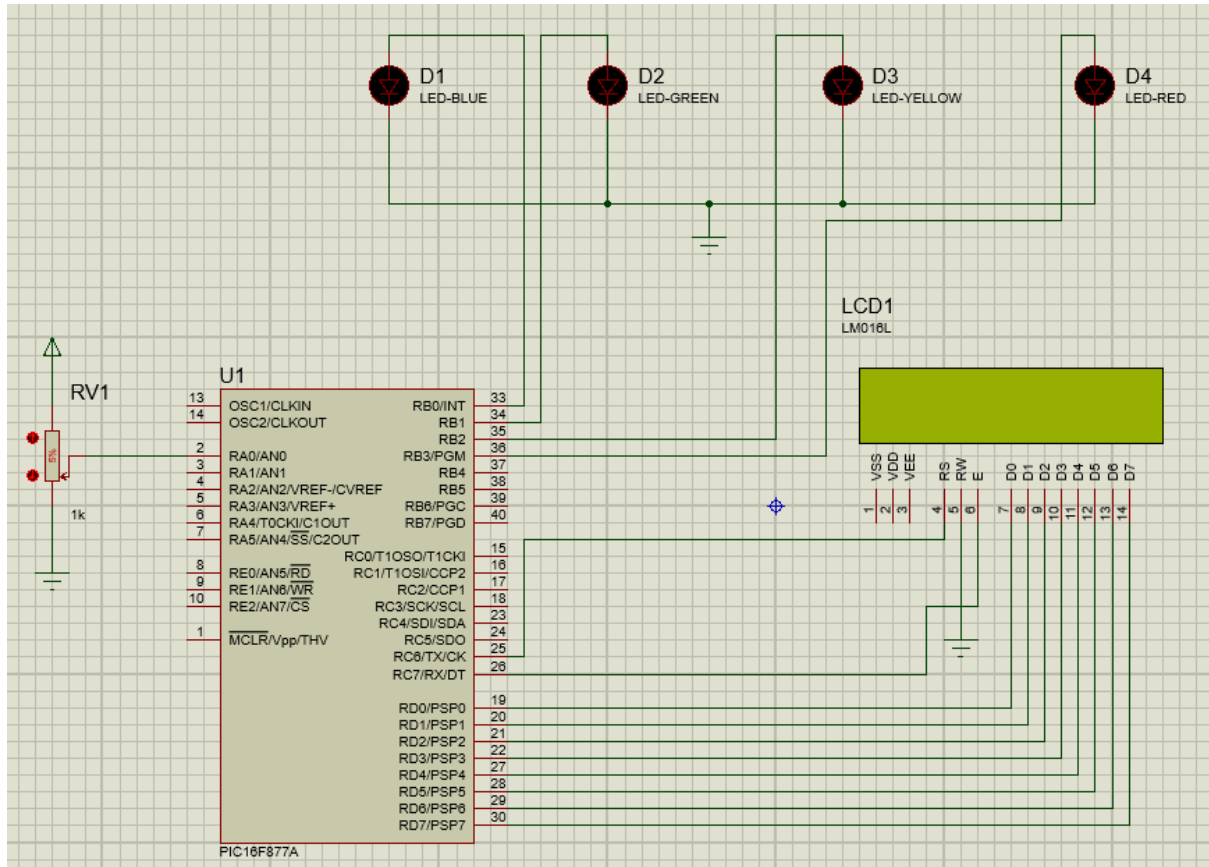
```
if(RESET>=250)  
{  
LED1=1;  
}  
else  
{  
LED1=0;  
}
```

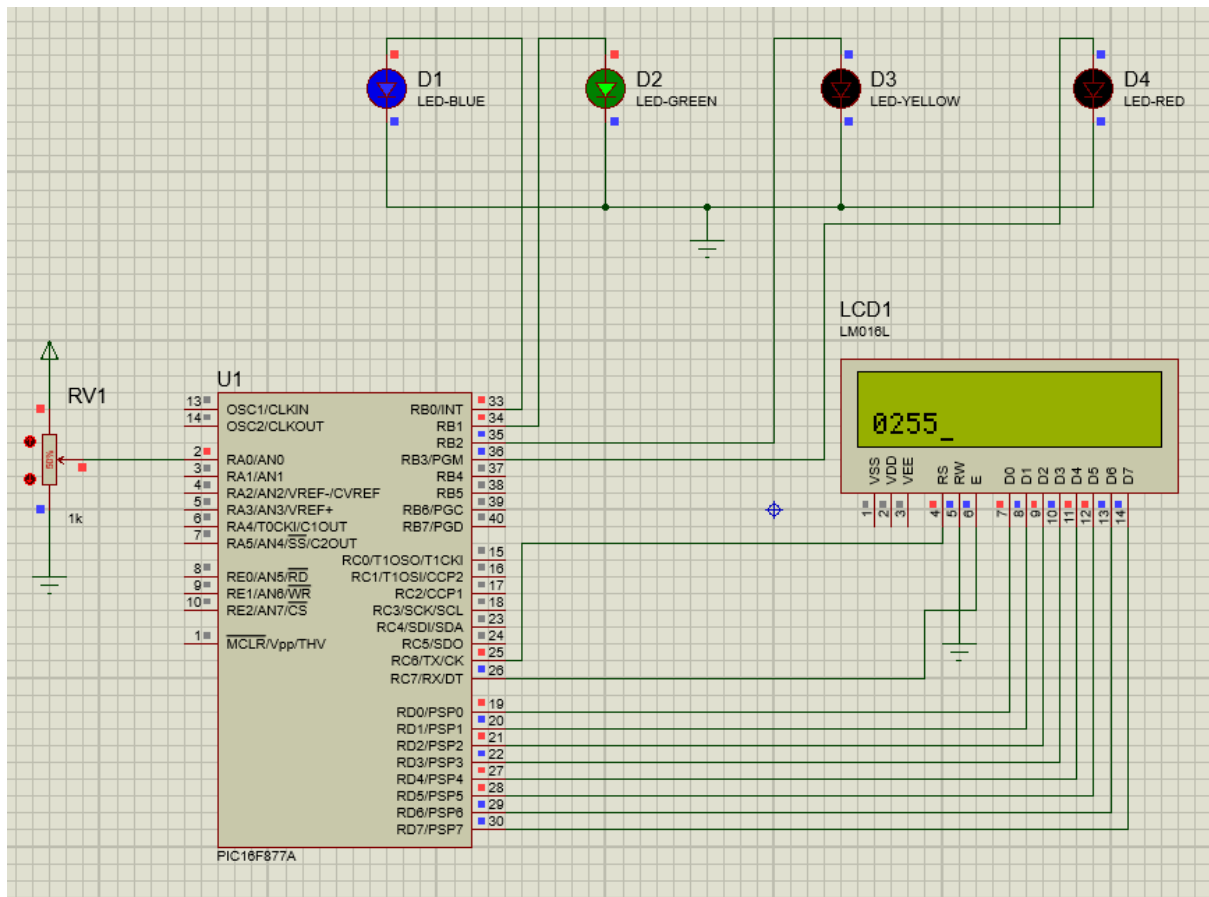
```
if(RESET>=500)  
{  
LED2=1;
```

```
}  
else  
{  
    LED2=0;  
}  
  
if(RESET>=750)  
{  
    LED3=1;  
}  
else  
{  
    LED3=0;  
}  
  
if(RESET>=1000)  
{  
    LED4=1;  
}  
else  
{  
    LED4=0;  
}  
  
}  
}  
  
void display(char x,int y)  
{  
    LCD=x;  
    RS=y;
```

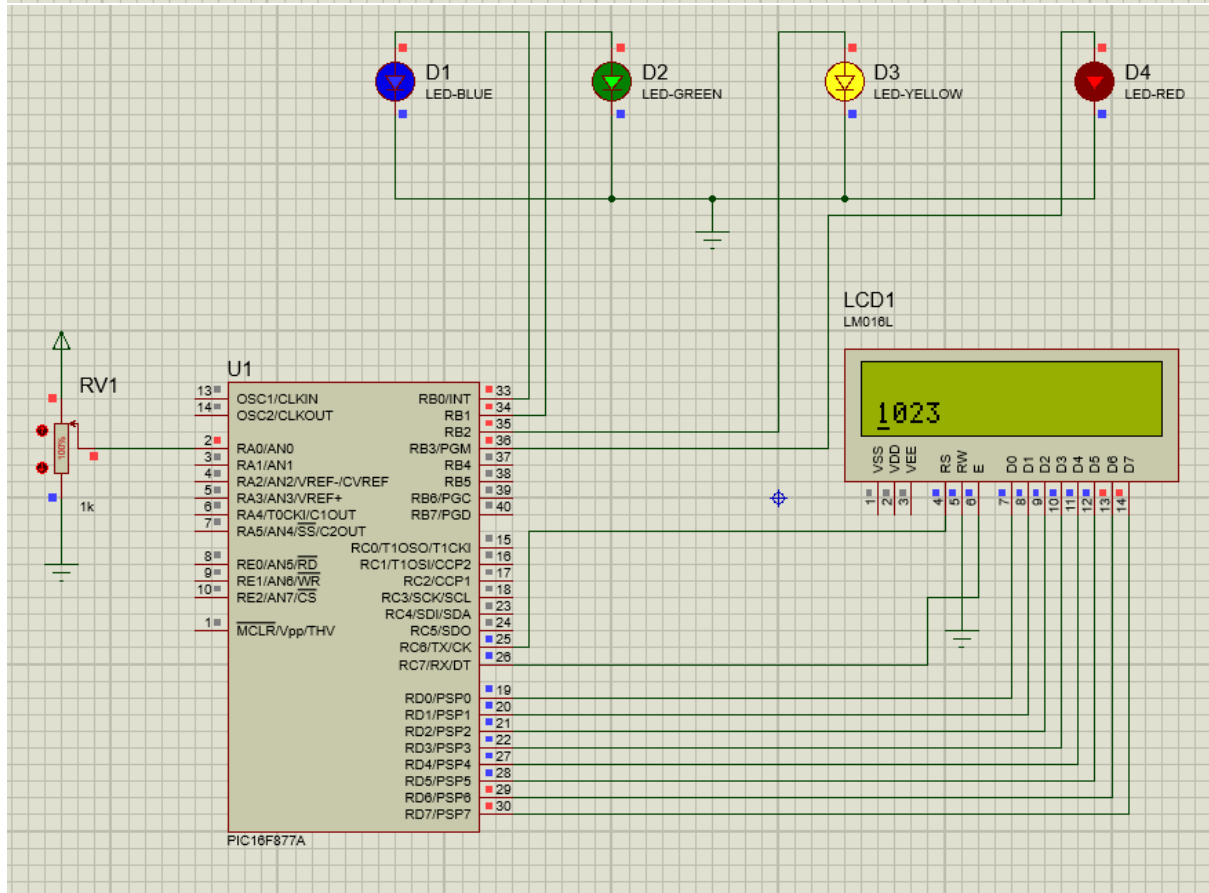
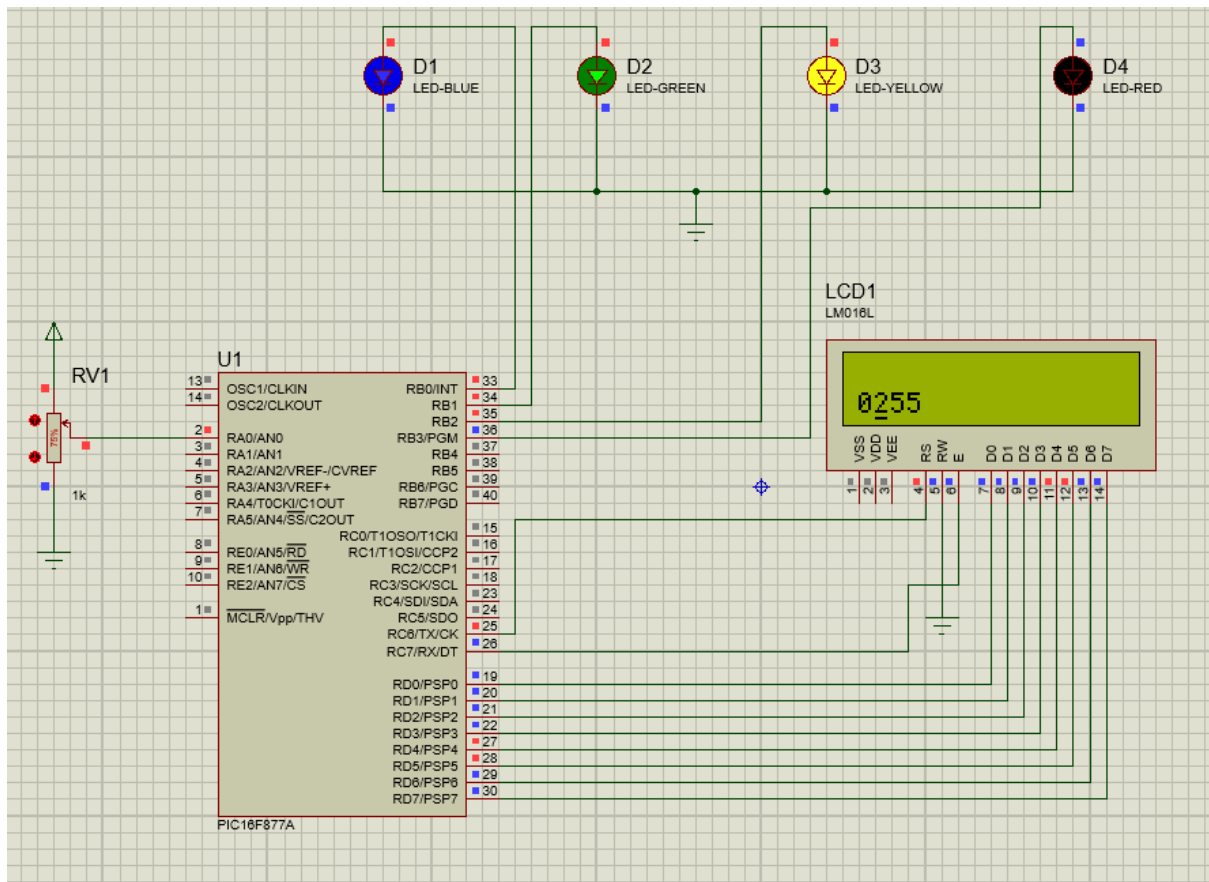
```
EN=1;  
delay_ms(3);  
EN=0;  
delay_ms(3);  
}
```

**SIMULATED OUTPUT:**









14). Write an embedded C Program for UART-TX(transmitter) . [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).

```
#include <UART1.h>
```

```
#byte SPBRG=0x99
```

```
#byte TXREG=0x19
```

```
#byte RCREG=0x1A
```

```
#bit TXIF=0x0c.4
```

```
#bit RCIF=0x0c.5
```

```
#byte TXSTA=0x98
```

```
#byte RCSTA=0x18
```

```
void transmit(char a)
```

```
{
```

```
TXREG=a;
```

```
while(TXIF==0);
```

```
TXIF=0;
```

```
}
```

```
void main()
```

```
{
```

```
int i;
```

```
char data[]="QISCET_ECE";
```

```
TXSTA=0x26;
```

```
RCSTA=0x90;
```

```
SPBRG=129;
```

```
while(1)
```

```
{
```

```

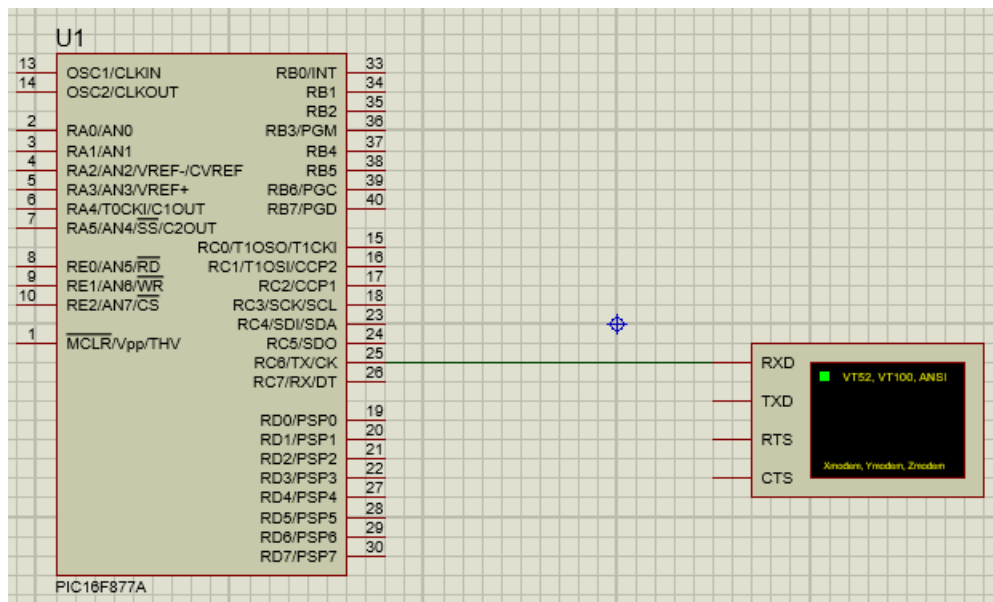
for(i=0;data[i]!=0;i++)
{
transmit(data[i]);
}

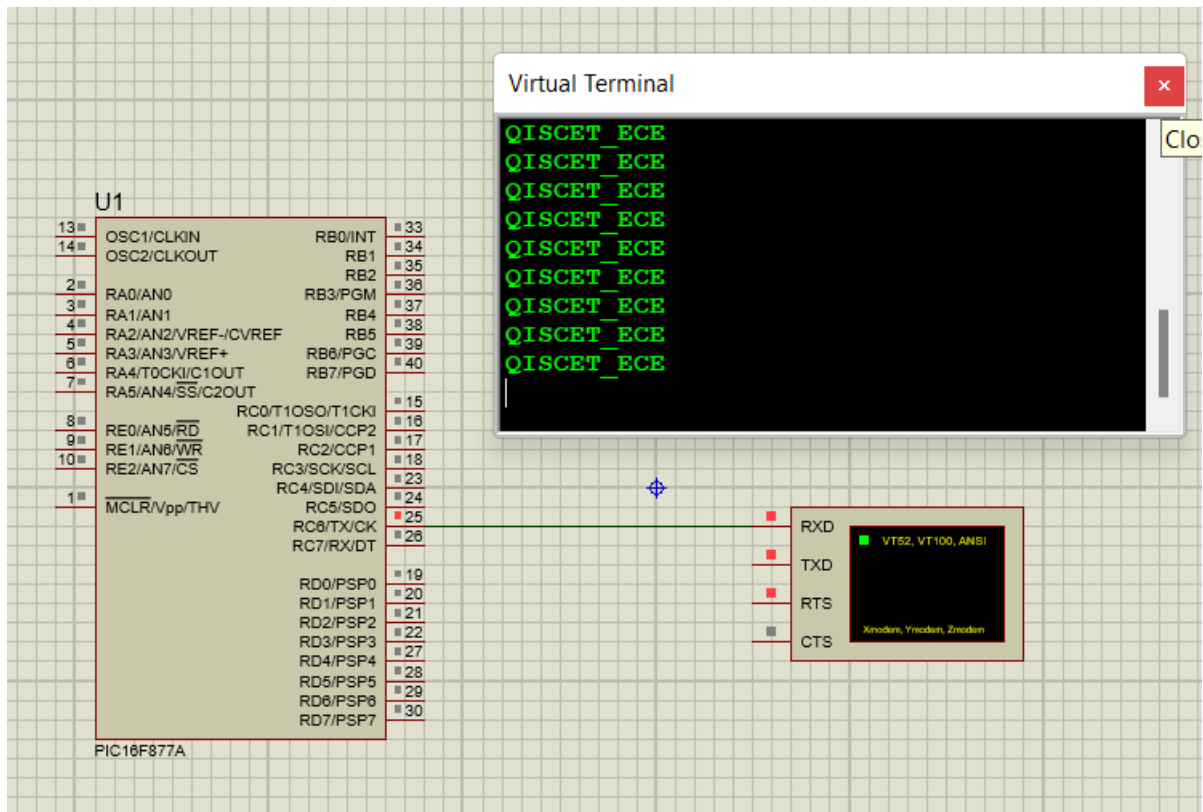
transmit('\r');

delay_ms(500);
}
}

```

### SIMULATED OUTPUT:





15). Write an embedded C Program for UART-RX(Receiver). [ complete the simulation? ] / execute this using proteus 8 Professional Software ?

A).