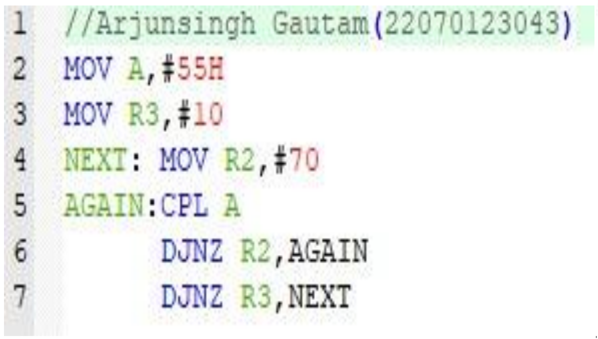
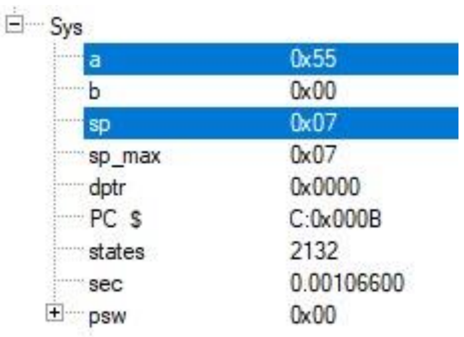
Exp-5



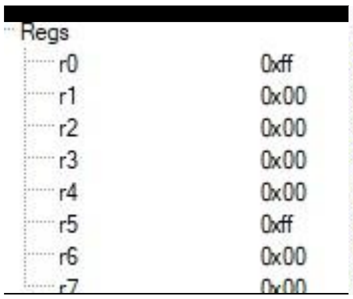
```
//Arjunsingh Gautam(22070123043)
MOV A,#55H
MOV R3,#10
NEXT: MOV R2,#70
AGAIN:CPL A
      DJNZ R2,AGAIN
      DJNZ R3,NEXT
```



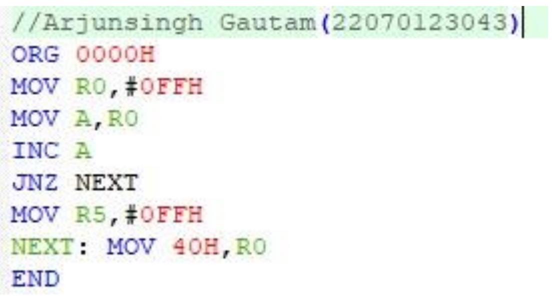
Sys	
a	0x55
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC $	C:0x000B
states	2132
sec	0.00106600
psw	0x00

Load ACC with 55H

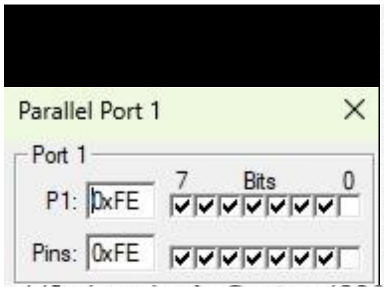
Complement ACC 700 times



Regs	
r0	0xff
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0xff
r6	0x00
r7	0x00



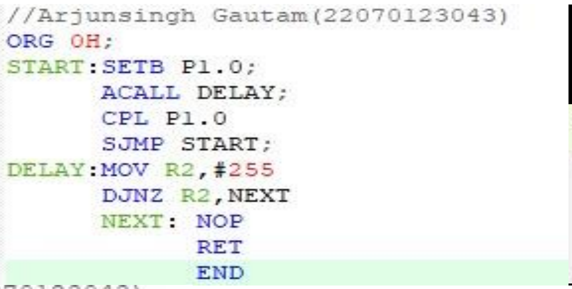
```
//Arjunsingh Gautam(22070123043)
ORG 0000H
MOV R0,#0FFH
MOV A,R0
INC A
JNZ NEXT
MOV R5,#0FFH
NEXT: MOV 40H,R0
END
```



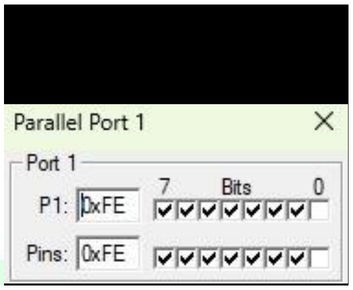
Parallel Port 1

Port 1

		7 Bits 0
P1:	0xFE	✓✓✓✓✓✓✓☐
Pins:	0xFE	✓✓✓✓✓✓✓☐



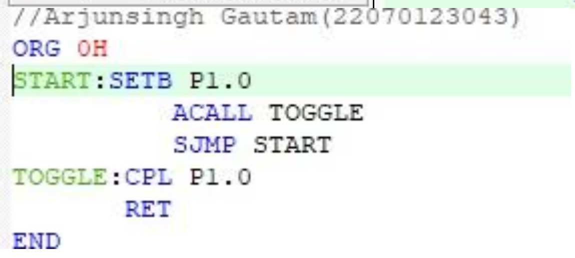
```
//Arjunsingh Gautam(22070123043)
ORG 0H;
START:SETB P1.0;
      ACALL DELAY;
      CPL P1.0
      SJMP START;
DELAY:MOV R2,#255
      DJNZ R2,NEXT
      NEXT: NOP
            RET
            END
```



Parallel Port 1

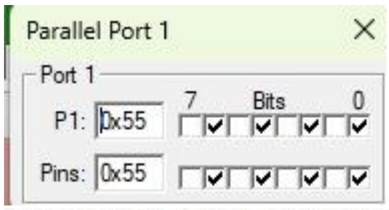
Port 1

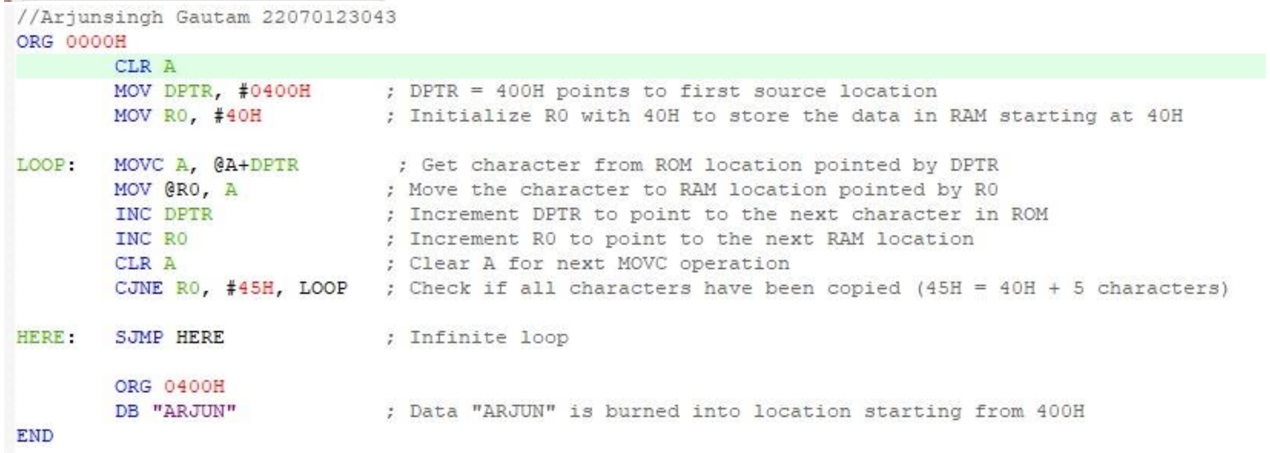
		7 Bits 0
P1:	0xFE	✓✓✓✓✓✓✓☐
Pins:	0xFE	✓✓✓✓✓✓✓☐

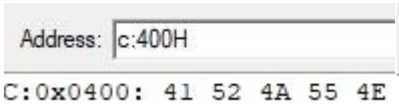


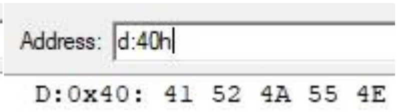
```
//Arjunsingh Gautam(22070123043)
ORG 0H
START:SETB P1.0
          ACALL TOGGLE
          SJMP START
TOGGLE:CPL P1.0
       RET
END
```

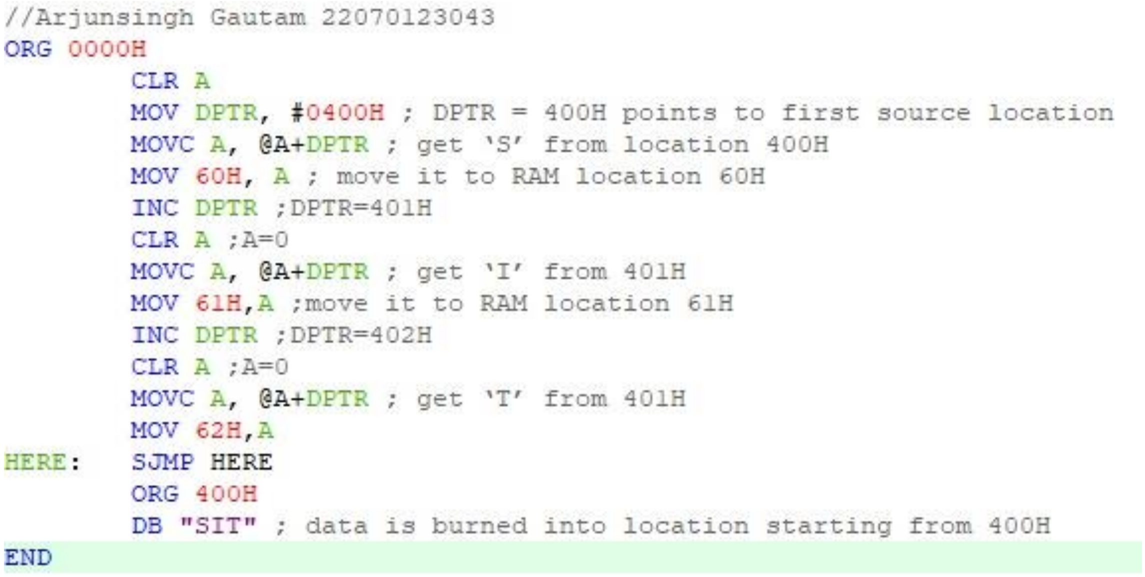
Exp6:

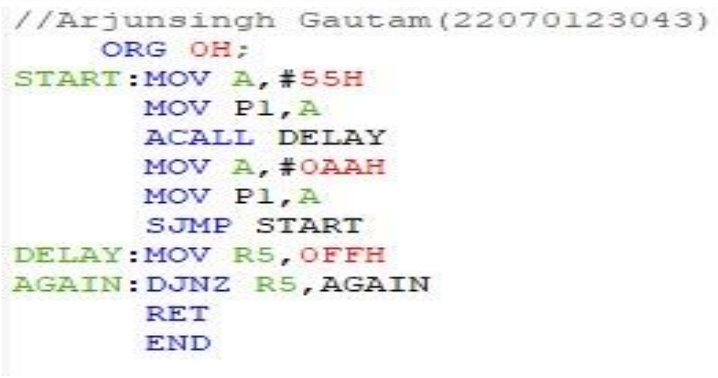

```
Parallel Port 1
Port 1
        7    Bits    0
P1:   0x55    ☐☑☐☑☐☑☐☑
Pins: 0x55    ☐☑☐☑☐☑☐☑
```

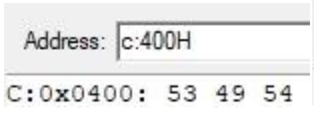

```
//Arjunsingh Gautam 22070123043
ORG 0000H
        CLR A
        MOV DPTR, #0400H      ; DPTR = 400H points to first source location
        MOV R0, #40H          ; Initialize R0 with 40H to store the data in RAM starting at 40H

LOOP:   MOVC A, @A+DPTR        ; Get character from ROM location pointed by DPTR
        MOV @R0, A            ; Move the character to RAM location pointed by R0
        INC DPTR              ; Increment DPTR to point to the next character in ROM
        INC R0                ; Increment R0 to point to the next RAM location
        CLR A                 ; Clear A for next MOVC operation
        CJNE R0, #45H, LOOP   ; Check if all characters have been copied (45H = 40H + 5 characters)

HERE:   SJMP HERE             ; Infinite loop

        ORG 0400H
        DB "ARJUN"            ; Data "ARJUN" is burned into location starting from 400H
END
```

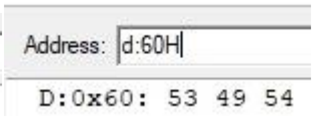

```
Address: c:400H
C:0x0400: 41 52 4A 55 4E
```


```
Address: d:40h
D:0x40: 41 52 4A 55 4E
```

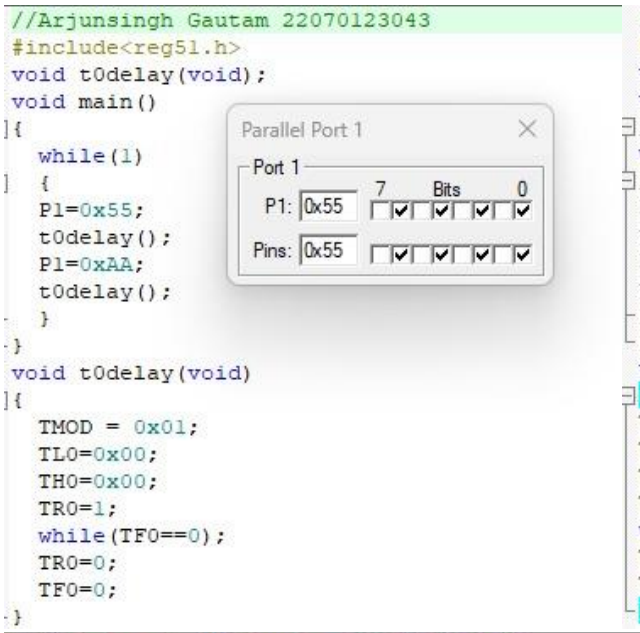

```
//Arjunsingh Gautam 22070123043
ORG 0000H
        CLR A
        MOV DPTR, #0400H ; DPTR = 400H points to first source location
        MOVC A, @A+DPTR ; get 'S' from location 400H
        MOV 60H, A ; move it to RAM location 60H
        INC DPTR ;DPTR=401H
        CLR A ;A=0
        MOVC A, @A+DPTR ; get 'I' from 401H
        MOV 61H,A ;move it to RAM location 61H
        INC DPTR ;DPTR=402H
        CLR A ;A=0
        MOVC A, @A+DPTR ; get 'T' from 401H
        MOV 62H,A
HERE:   SJMP HERE
        ORG 400H
        DB "SIT" ; data is burned into location starting from 400H
END
```


```
//Arjunsingh Gautam(22070123043)
      ORG OH;
START:MOV A,#55H
      MOV P1,A
      ACALL DELAY
      MOV A,#0AAH
      MOV P1,A
      SJMP START
DELAY:MOV R5,OFFH
AGAIN:DJNZ R5,AGAIN
      RET
      END
```


```
Address: c:400H
C:0x0400: 53 49 54
```

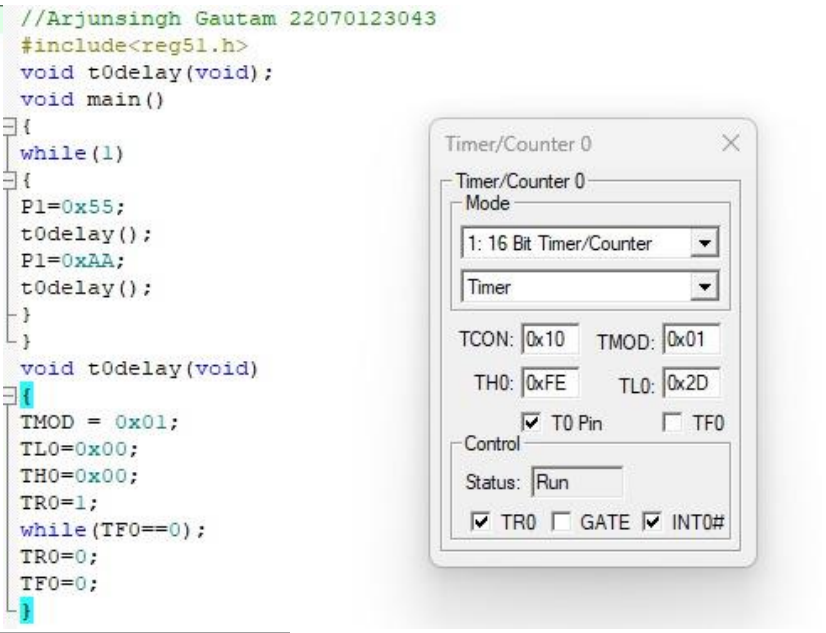

```
Address: d:60H
D:0x60: 53 49 54
```

Exp-7



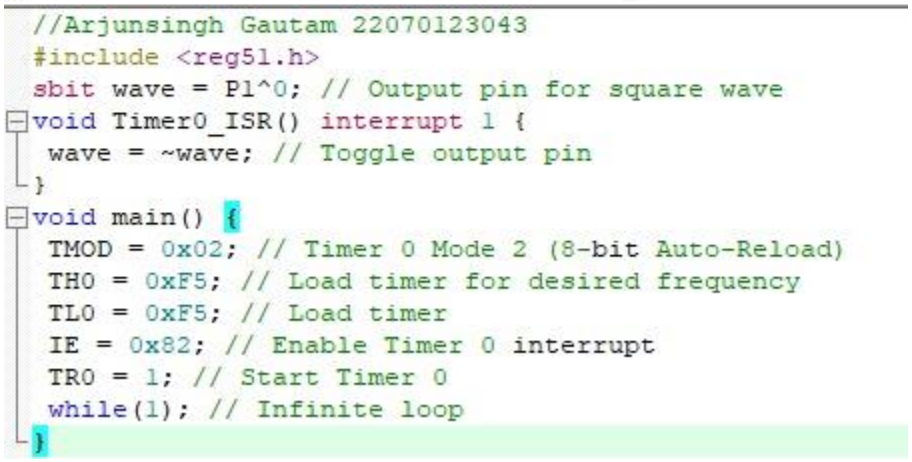
```
//Arjunsingh Gautam 22070123043
#include<reg51.h>
void t0delay(void);
void main()
{
  while(1)
  {
  P1=0x55;
  t0delay();
  P1=0xAA;
  t0delay();
  }
}
void t0delay(void)
{
  TMOD = 0x01;
  TL0=0x00;
  TH0=0x00;
  TR0=1;
  while(TF0==0);
  TR0=0;
  TF0=0;
}
```

Parallel Port 1
Port 1
P1: 0x55 — Bits 7 … 0: ☐ ✓ ☐ ✓ ☐ ✓ ☐ ✓
Pins: 0x55 — ☐ ✓ ☐ ✓ ☐ ✓ ☐ ✓



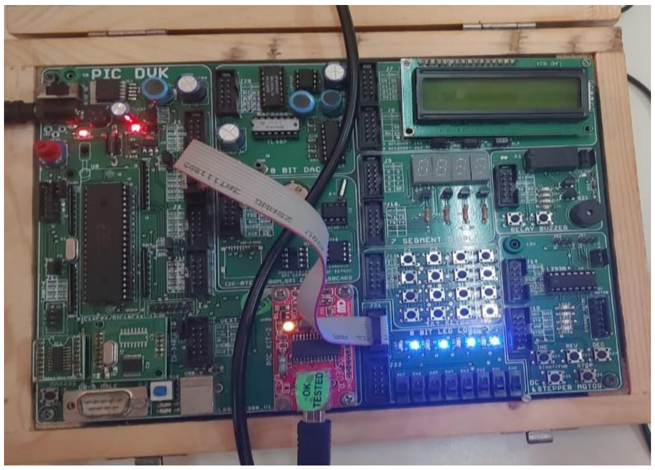
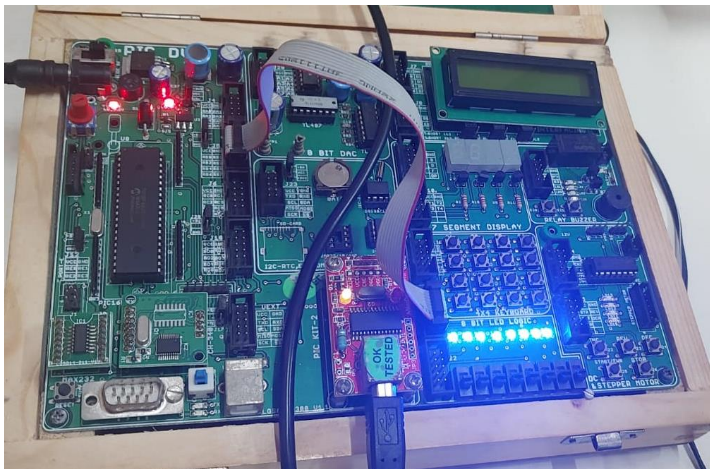
```
//Arjunsingh Gautam 22070123043
#include<reg51.h>
void t0delay(void);
void main()
{
while(1)
{
P1=0x55;
t0delay();
P1=0xAA;
t0delay();
}
}
void t0delay(void)
{
TMOD = 0x01;
TL0=0x00;
TH0=0x00;
TR0=1;
while(TF0==0);
TR0=0;
TF0=0;
}
```

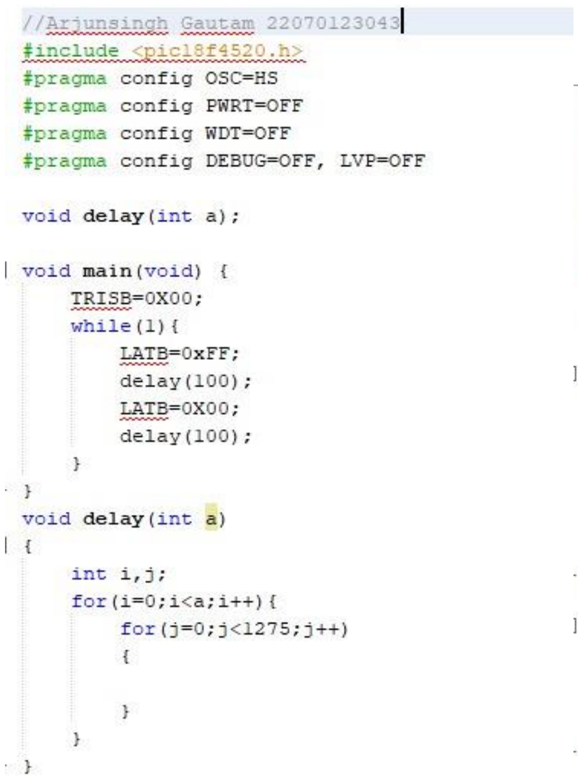
Timer/Counter 0
Timer/Counter 0
Mode: 1: 16 Bit Timer/Counter
Timer
TCON: 0x10 TMOD: 0x01
TH0: 0xFE TL0: 0x2D
[✓] T0 Pin [ ] TF0
Control
Status: Run
[✓] TR0 [ ] GATE [✓] INT0#



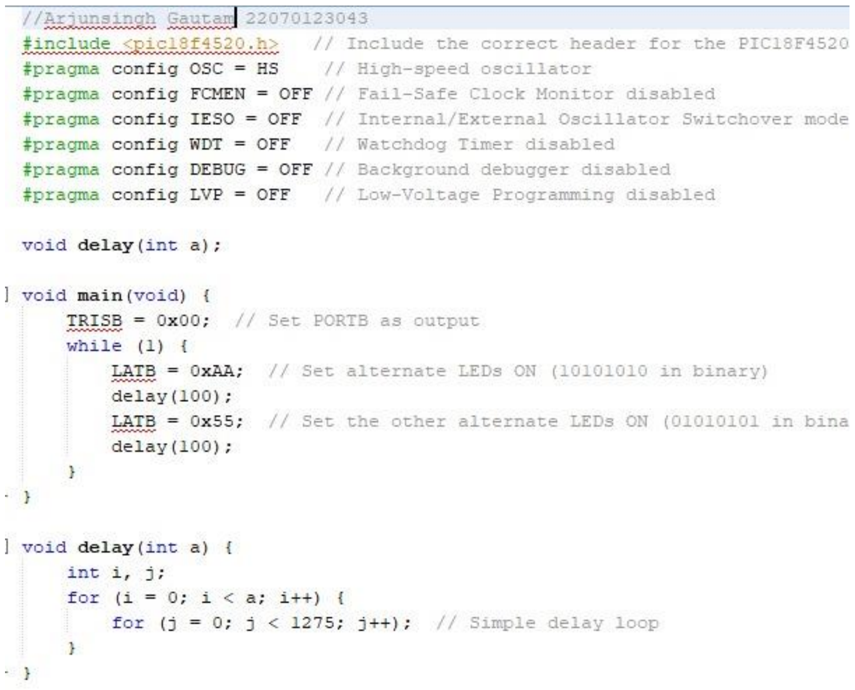
```
//Arjunsingh Gautam 22070123043
#include <reg51.h>
sbit wave = P1^0; // Output pin for square wave
void Timer0_ISR() interrupt 1 {
 wave = ~wave; // Toggle output pin
}
void main() {
 TMOD = 0x02; // Timer 0 Mode 2 (8-bit Auto-Reload)
 TH0 = 0xF5; // Load timer for desired frequency
 TL0 = 0xF5; // Load timer
 IE = 0x82; // Enable Timer 0 interrupt
 TR0 = 1; // Start Timer 0
 while(1); // Infinite loop
}
```

Exp-8

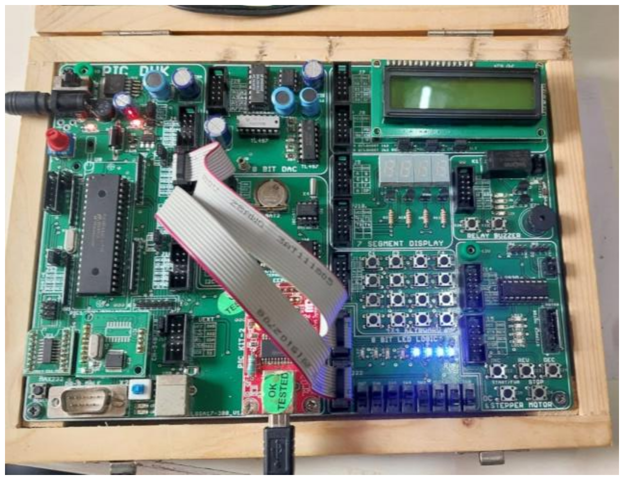


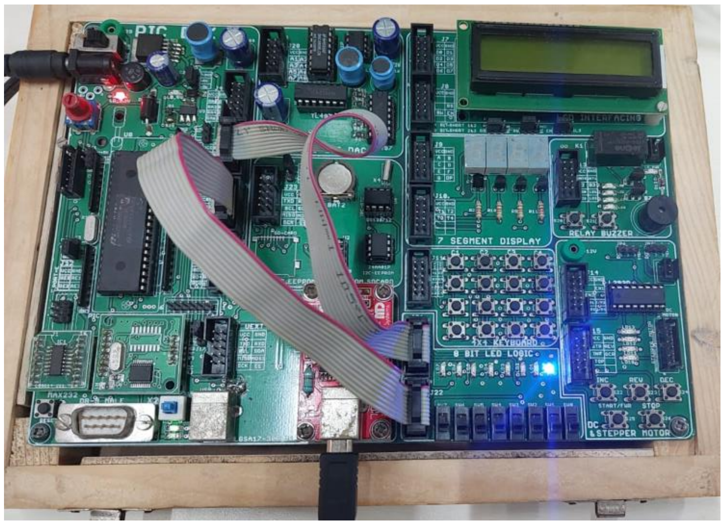
```
//Arjunsingh Gautam 22070123043
#include <pic18f4520.h>
#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF, LVP=OFF

void delay(int a);

void main(void) {
    TRISB=0X00;
    while(1){
        LATB=0xFF;
        delay(100);
        LATB=0X00;
        delay(100);
    }
}
void delay(int a)
{
    int i,j;
    for(i=0;i<a;i++){
        for(j=0;j<1275;j++)
        {

        }
    }
}
```

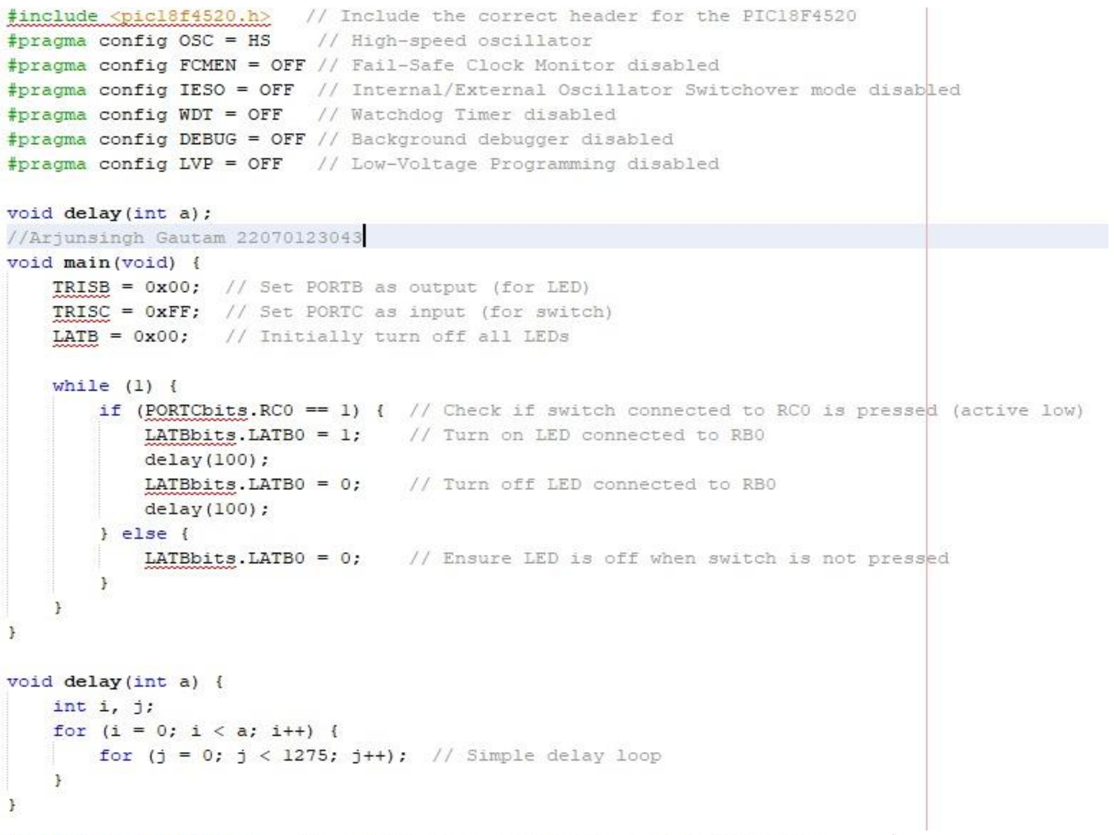


```
//Arjunsingh Gautam 22070123043
#include <pic18f4520.h>   // Include the correct header for the PIC18F4520
#pragma config OSC = HS    // High-speed oscillator
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor disabled
#pragma config IESO = OFF  // Internal/External Oscillator Switchover mode
#pragma config WDT = OFF   // Watchdog Timer disabled
#pragma config DEBUG = OFF // Background debugger disabled
#pragma config LVP = OFF   // Low-Voltage Programming disabled

void delay(int a);

void main(void) {
    TRISB = 0x00;  // Set PORTB as output
    while (1) {
        LATB = 0xAA;  // Set alternate LEDs ON (10101010 in binary)
        delay(100);
        LATB = 0x55;  // Set the other alternate LEDs ON (01010101 in bina
        delay(100);
    }
}

void delay(int a) {
    int i, j;
    for (i = 0; i < a; i++) {
        for (j = 0; j < 1275; j++);  // Simple delay loop
    }
}
```

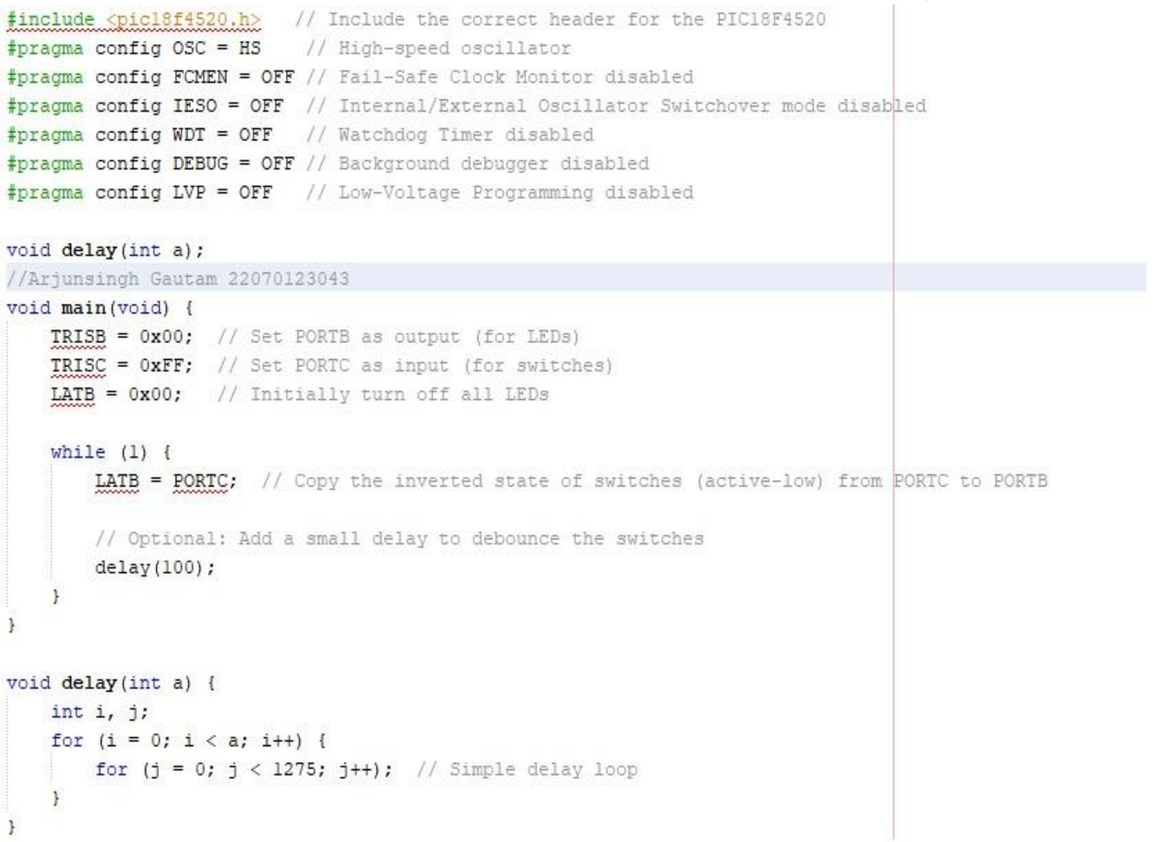
Exp-9



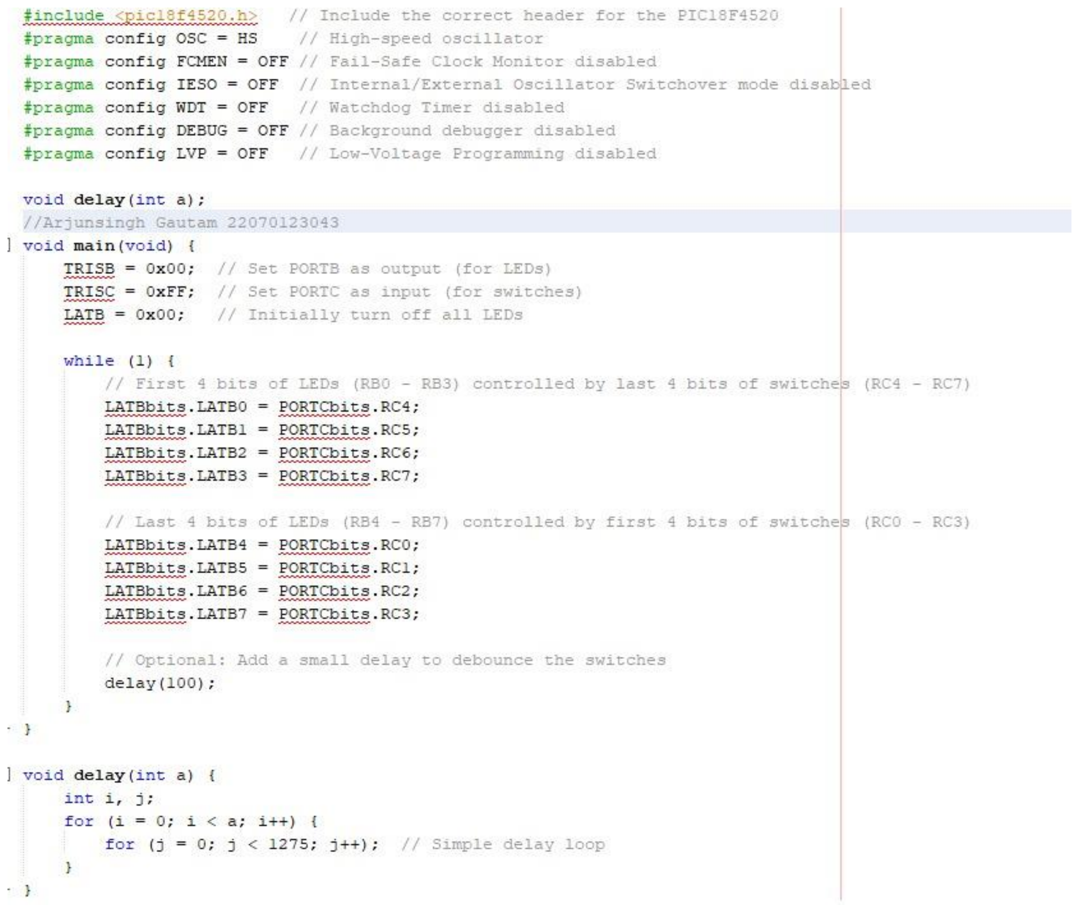




```c
#include <pic18f4520.h>   // Include the correct header for the PIC18F4520
#pragma config OSC = HS    // High-speed oscillator
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor disabled
#pragma config IESO = OFF  // Internal/External Oscillator Switchover mode disabled
#pragma config WDT = OFF   // Watchdog Timer disabled
#pragma config DEBUG = OFF // Background debugger disabled
#pragma config LVP = OFF   // Low-Voltage Programming disabled

void delay(int a);
//Arjunsingh Gautam 22070123043
void main(void) {
    TRISB = 0x00;  // Set PORTB as output (for LED)
    TRISC = 0xFF;  // Set PORTC as input (for switch)
    LATB = 0x00;   // Initially turn off all LEDs

    while (1) {
        if (PORTCbits.RC0 == 1) {  // Check if switch connected to RC0 is pressed (active low)
            LATBbits.LATB0 = 1;    // Turn on LED connected to RB0
            delay(100);
            LATBbits.LATB0 = 0;    // Turn off LED connected to RB0
            delay(100);
        } else {
            LATBbits.LATB0 = 0;    // Ensure LED is off when switch is not pressed
        }
    }
}

void delay(int a) {
    int i, j;
    for (i = 0; i < a; i++) {
        for (j = 0; j < 1275; j++);  // Simple delay loop
    }
}
```

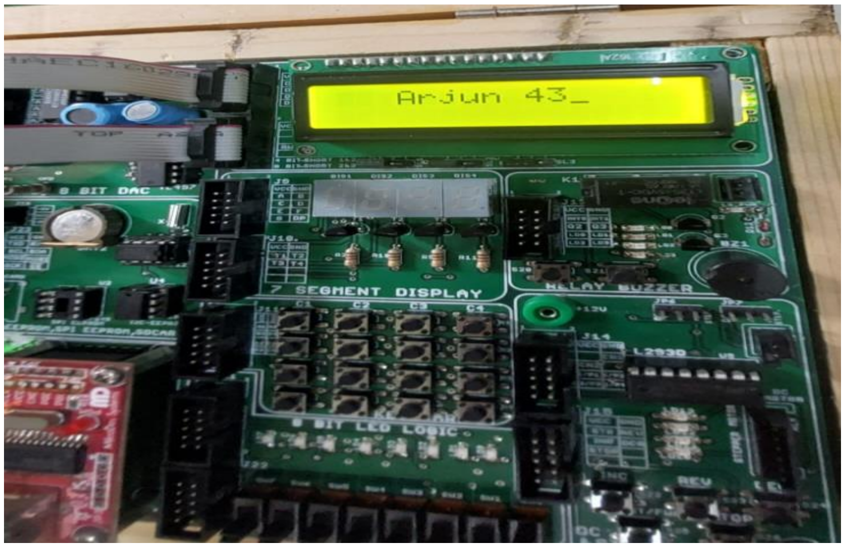


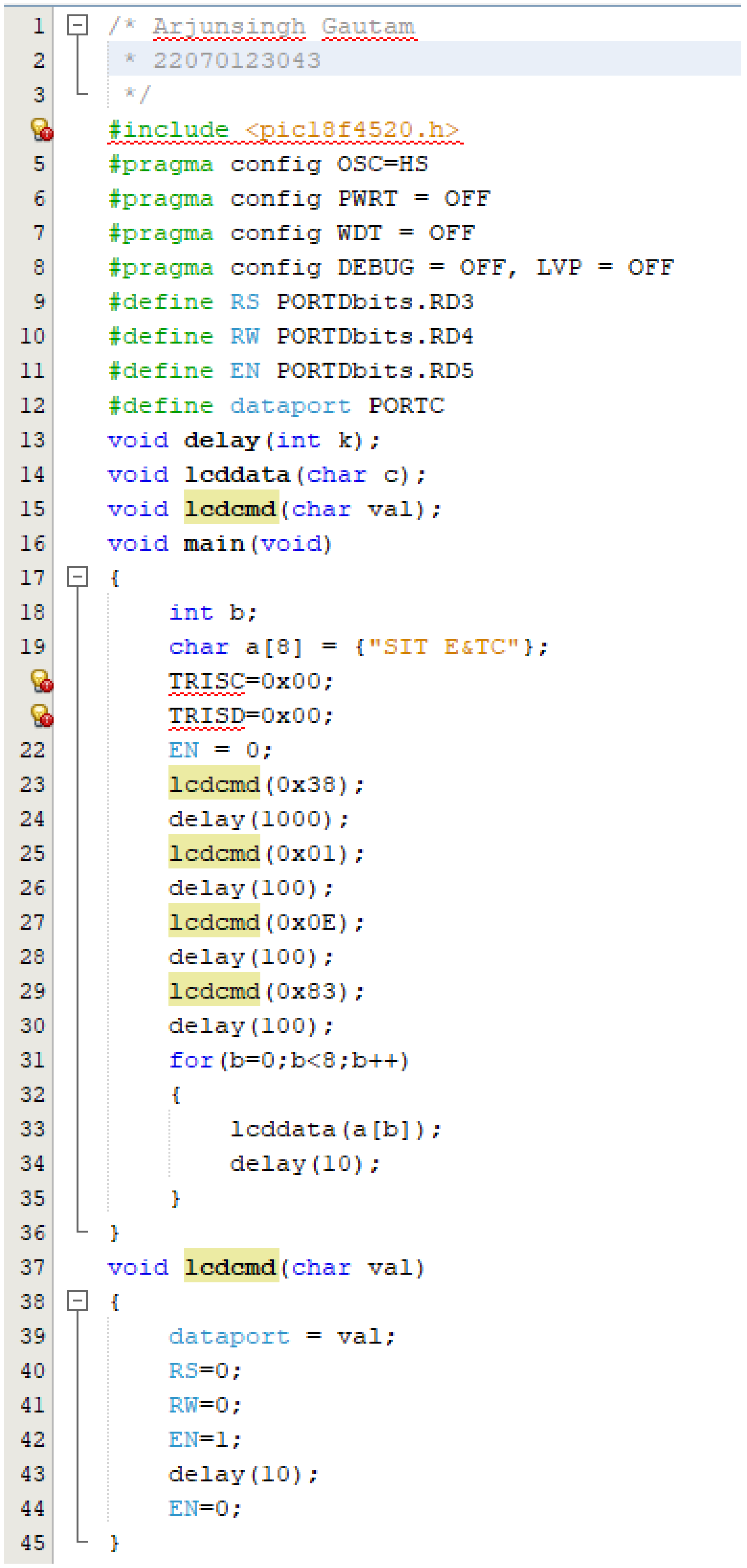
```c
#include <pic18f4520.h>   // Include the correct header for the PIC18F4520
#pragma config OSC = HS    // High-speed oscillator
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor disabled
#pragma config IESO = OFF  // Internal/External Oscillator Switchover mode disabled
#pragma config WDT = OFF   // Watchdog Timer disabled
#pragma config DEBUG = OFF // Background debugger disabled
#pragma config LVP = OFF   // Low-Voltage Programming disabled

void delay(int a);
//Arjunsingh Gautam 22070123043
void main(void) {
    TRISB = 0x00;  // Set PORTB as output (for LEDs)
    TRISC = 0xFF;  // Set PORTC as input (for switches)
    LATB = 0x00;   // Initially turn off all LEDs

    while (1) {
        LATB = PORTC;  // Copy the inverted state of switches (active-low) from PORTC to PORTB

        // Optional: Add a small delay to debounce the switches
        delay(100);
    }
}

void delay(int a) {
    int i, j;
    for (i = 0; i < a; i++) {
        for (j = 0; j < 1275; j++);  // Simple delay loop
    }
}
```



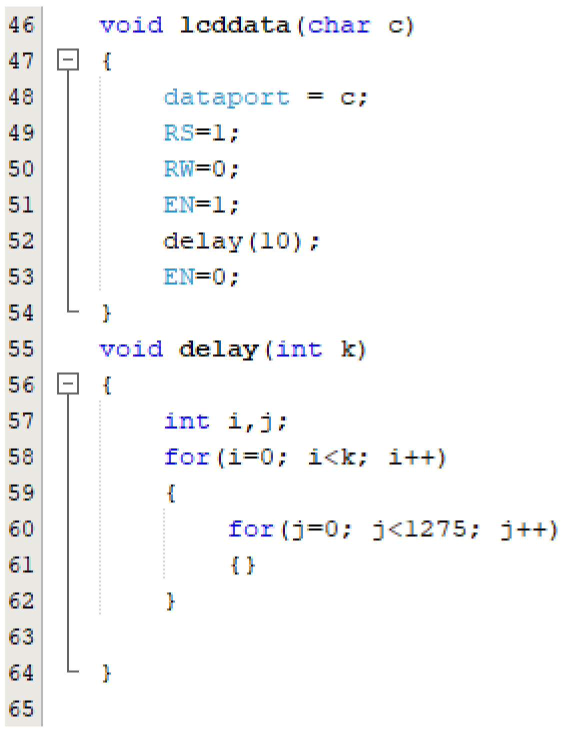
```
#include <pic18f4520.h>   // Include the correct header for the PIC18F4520
#pragma config OSC = HS    // High-speed oscillator
#pragma config FCMEN = OFF // Fail-Safe Clock Monitor disabled
#pragma config IESO = OFF  // Internal/External Oscillator Switchover mode disabled
#pragma config WDT = OFF   // Watchdog Timer disabled
#pragma config DEBUG = OFF // Background debugger disabled
#pragma config LVP = OFF   // Low-Voltage Programming disabled

void delay(int a);
//Arjunsingh Gautam 22070123043
void main(void) {
    TRISB = 0x00;  // Set PORTB as output (for LEDs)
    TRISC = 0xFF;  // Set PORTC as input (for switches)
    LATB = 0x00;   // Initially turn off all LEDs

    while (1) {
        // First 4 bits of LEDs (RB0 - RB3) controlled by last 4 bits of switches (RC4 - RC7)
        LATBbits.LATB0 = PORTCbits.RC4;
        LATBbits.LATB1 = PORTCbits.RC5;
        LATBbits.LATB2 = PORTCbits.RC6;
        LATBbits.LATB3 = PORTCbits.RC7;

        // Last 4 bits of LEDs (RB4 - RB7) controlled by first 4 bits of switches (RC0 - RC3)
        LATBbits.LATB4 = PORTCbits.RC0;
        LATBbits.LATB5 = PORTCbits.RC1;
        LATBbits.LATB6 = PORTCbits.RC2;
        LATBbits.LATB7 = PORTCbits.RC3;

        // Optional: Add a small delay to debounce the switches
        delay(100);
    }
}

void delay(int a) {
    int i, j;
    for (i = 0; i < a; i++) {
        for (j = 0; j < 1275; j++);  // Simple delay loop
    }
}
```

Exp-10



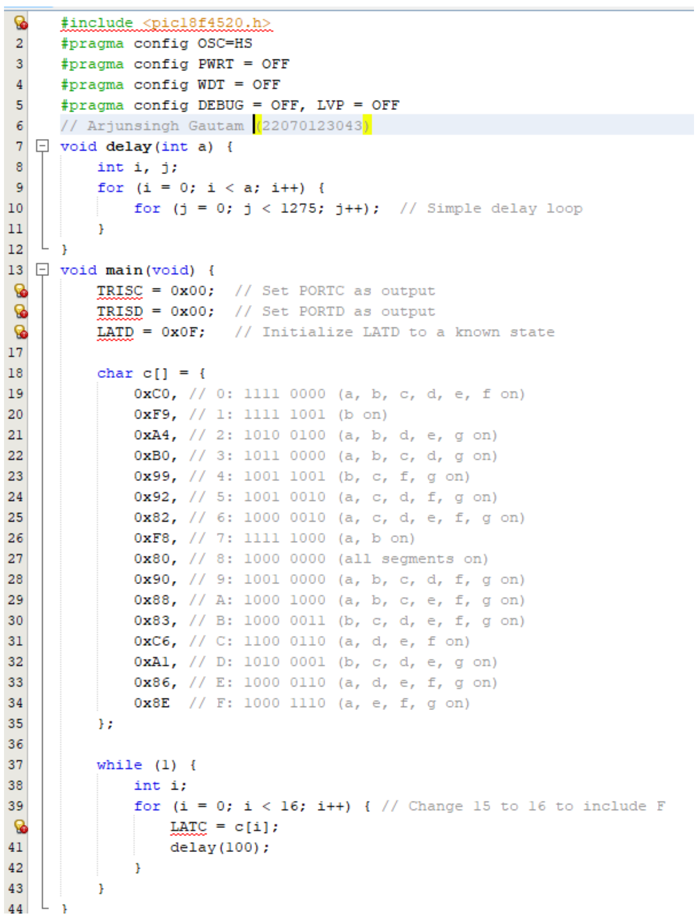


```c
/* Arjunsingh Gautam
 * 22070123043
 */
#include <pic18f4520.h>
#pragma config OSC=HS
#pragma config PWRT = OFF
#pragma config WDT = OFF
#pragma config DEBUG = OFF, LVP = OFF
#define RS PORTDbits.RD3
#define RW PORTDbits.RD4
#define EN PORTDbits.RD5
#define dataport PORTC
void delay(int k);
void lcddata(char c);
void lcdcmd(char val);
void main(void)
{
    int b;
    char a[8] = {"SIT E&TC"};
    TRISC=0x00;
    TRISD=0x00;
    EN = 0;
    lcdcmd(0x38);
    delay(1000);
    lcdcmd(0x01);
    delay(100);
    lcdcmd(0x0E);
    delay(100);
    lcdcmd(0x83);
    delay(100);
    for(b=0;b<8;b++)
    {
        lcddata(a[b]);
        delay(10);
    }
}
void lcdcmd(char val)
{
    dataport = val;
    RS=0;
    RW=0;
    EN=1;
    delay(10);
    EN=0;
}
```

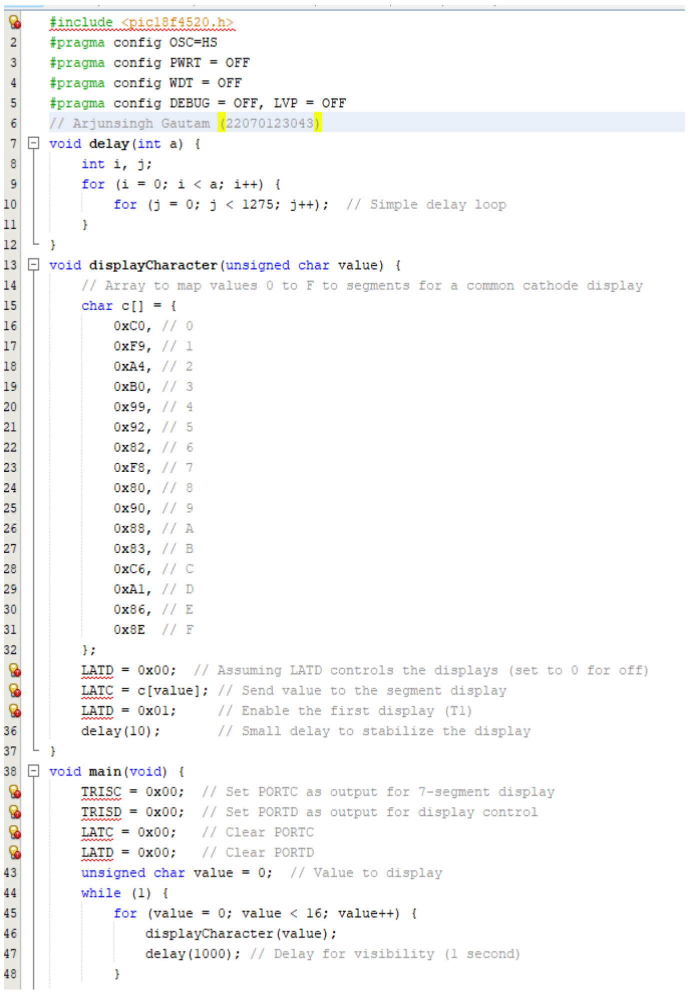


```c
void lcddata(char c)
{
    dataport = c;
    RS=1;
    RW=0;
    EN=1;
    delay(10);
    EN=0;
}
void delay(int k)
{
    int i,j;
    for(i=0; i<k; i++)
    {
        for(j=0; j<1275; j++)
        {}
    }

}

```

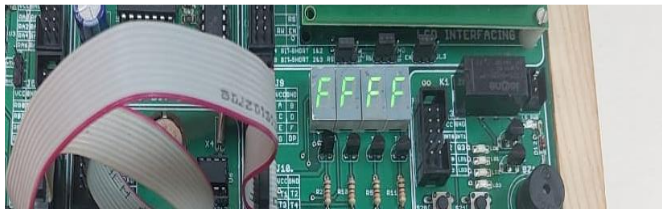
Exp-11

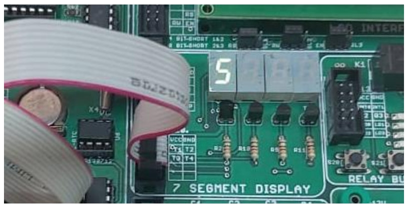


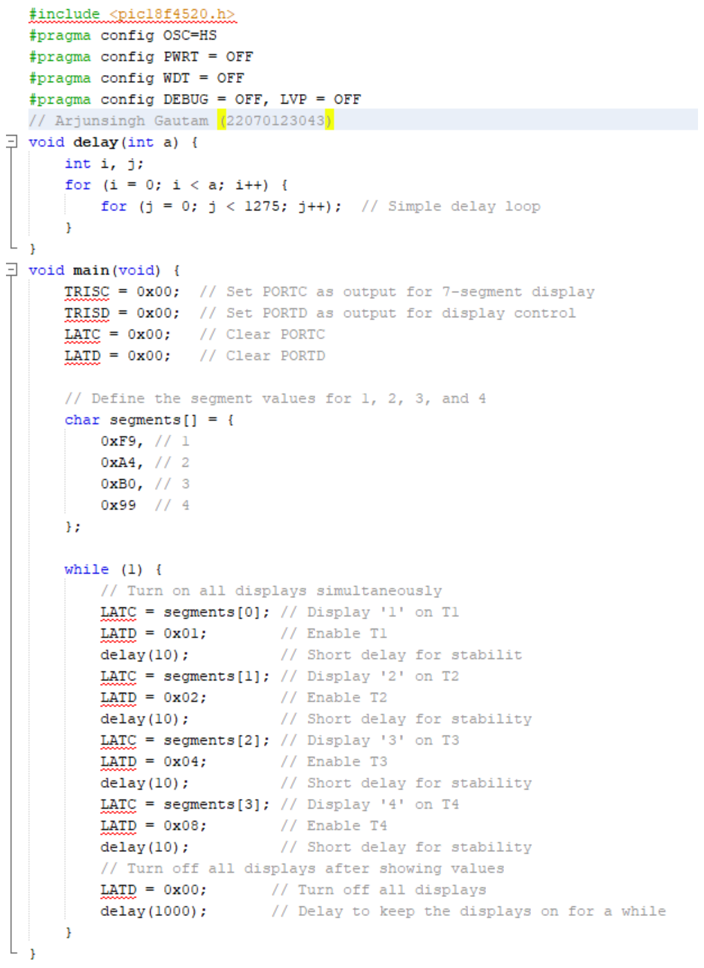
```c
#include <pic18f4520.h>
#pragma config OSC=HS
#pragma config PWRT = OFF
#pragma config WDT = OFF
#pragma config DEBUG = OFF, LVP = OFF
// Arjunsingh Gautam (22070123043)
void delay(int a) {
    int i, j;
    for (i = 0; i < a; i++) {
        for (j = 0; j < 1275; j++);  // Simple delay loop
    }
}
void main(void) {
    TRISC = 0x00;  // Set PORTC as output
    TRISD = 0x00;  // Set PORTD as output
    LATD = 0x0F;   // Initialize LATD to a known state

    char c[] = {
        0xC0, // 0: 1111 0000 (a, b, c, d, e, f on)
        0xF9, // 1: 1111 1001 (b on)
        0xA4, // 2: 1010 0100 (a, b, d, e, g on)
        0xB0, // 3: 1011 0000 (a, b, c, d, g on)
        0x99, // 4: 1001 1001 (b, c, f, g on)
        0x92, // 5: 1001 0010 (a, c, d, f, g on)
        0x82, // 6: 1000 0010 (a, c, d, e, f, g on)
        0xF8, // 7: 1111 1000 (a, b on)
        0x80, // 8: 1000 0000 (all segments on)
        0x90, // 9: 1001 0000 (a, b, c, d, f, g on)
        0x88, // A: 1000 1000 (a, b, c, e, f, g on)
        0x83, // B: 1000 0011 (b, c, d, e, f, g on)
        0xC6, // C: 1100 0110 (a, d, e, f on)
        0xA1, // D: 1010 0001 (b, c, d, e, g on)
        0x86, // E: 1000 0110 (a, d, e, f, g on)
        0x8E  // F: 1000 1110 (a, e, f, g on)
    };

    while (1) {
        int i;
        for (i = 0; i < 16; i++) { // Change 15 to 16 to include F
            LATC = c[i];
            delay(100);
        }
    }
}
```



```c
#include <pic18f4520.h>
#pragma config OSC=HS
#pragma config PWRT = OFF
#pragma config WDT = OFF
#pragma config DEBUG = OFF, LVP = OFF
// Arjunsingh Gautam (22070123043)
void delay(int a) {
    int i, j;
    for (i = 0; i < a; i++) {
        for (j = 0; j < 1275; j++);  // Simple delay loop
    }
}
void displayCharacter(unsigned char value) {
    // Array to map values 0 to F to segments for a common cathode display
    char c[] = {
        0xC0, // 0
        0xF9, // 1
        0xA4, // 2
        0xB0, // 3
        0x99, // 4
        0x92, // 5
        0x82, // 6
        0xF8, // 7
        0x80, // 8
        0x90, // 9
        0x88, // A
        0x83, // B
        0xC6, // C
        0xA1, // D
        0x86, // E
        0x8E  // F
    };
    LATD = 0x00;  // Assuming LATD controls the displays (set to 0 for off)
    LATC = c[value]; // Send value to the segment display
    LATD = 0x01;     // Enable the first display (T1)
    delay(10);       // Small delay to stabilize the display
}
void main(void) {
    TRISC = 0x00;  // Set PORTC as output for 7-segment display
    TRISD = 0x00;  // Set PORTD as output for display control
    LATC = 0x00;   // Clear PORTC
    LATD = 0x00;   // Clear PORTD
    unsigned char value = 0;  // Value to display
    while (1) {
        for (value = 0; value < 16; value++) {
            displayCharacter(value);
            delay(1000); // Delay for visibility (1 second)
        }
```

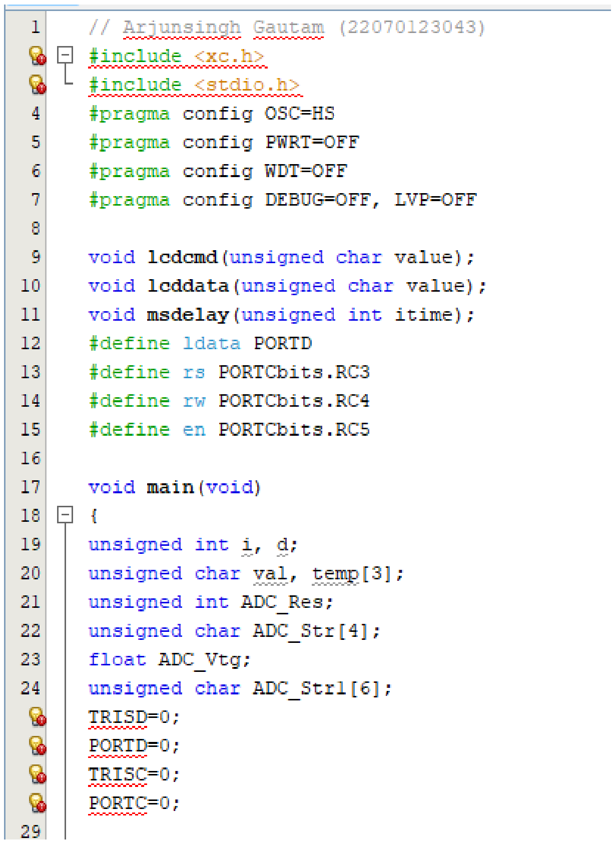




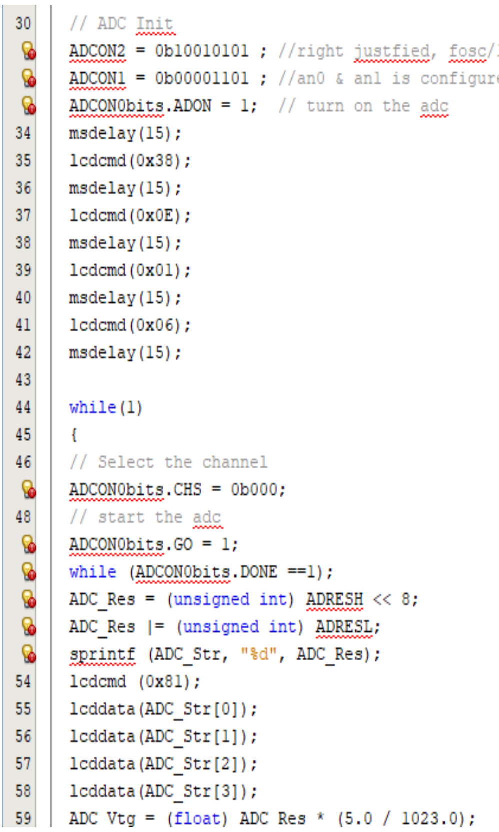


```c
#include <pic18f4520.h>
#pragma config OSC=HS
#pragma config PWRT = OFF
#pragma config WDT = OFF
#pragma config DEBUG = OFF, LVP = OFF
// Arjunsingh Gautam (22070123043)
void delay(int a) {
    int i, j;
    for (i = 0; i < a; i++) {
        for (j = 0; j < 1275; j++);  // Simple delay loop
    }
}
void main(void) {
    TRISC = 0x00;  // Set PORTC as output for 7-segment display
    TRISD = 0x00;  // Set PORTD as output for display control
    LATC = 0x00;   // Clear PORTC
    LATD = 0x00;   // Clear PORTD

    // Define the segment values for 1, 2, 3, and 4
    char segments[] = {
        0xF9, // 1
        0xA4, // 2
        0xB0, // 3
        0x99  // 4
    };

    while (1) {
        // Turn on all displays simultaneously
        LATC = segments[0]; // Display '1' on T1
        LATD = 0x01;        // Enable T1
        delay(10);          // Short delay for stabilit
        LATC = segments[1]; // Display '2' on T2
        LATD = 0x02;        // Enable T2
        delay(10);          // Short delay for stability
        LATC = segments[2]; // Display '3' on T3
        LATD = 0x04;        // Enable T3
        delay(10);          // Short delay for stability
        LATC = segments[3]; // Display '4' on T4
        LATD = 0x08;        // Enable T4
        delay(10);          // Short delay for stability
        // Turn off all displays after showing values
        LATD = 0x00;       // Turn off all displays
        delay(1000);       // Delay to keep the displays on for a while
    }
}
```

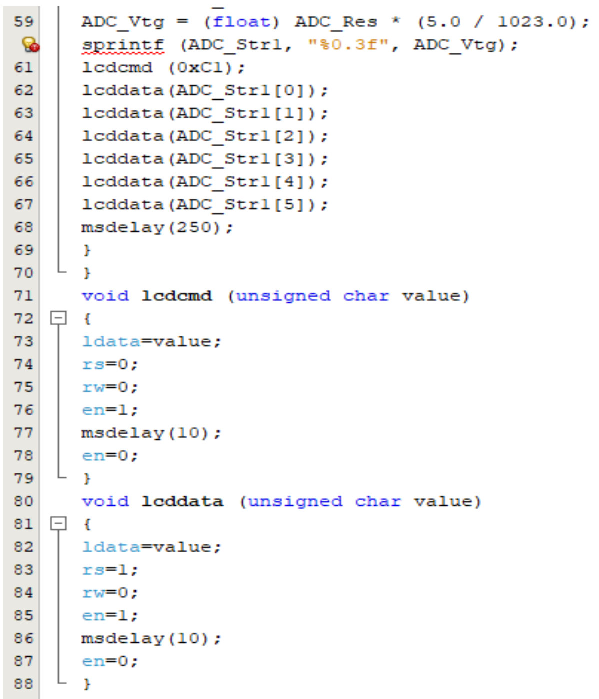
Exp-12



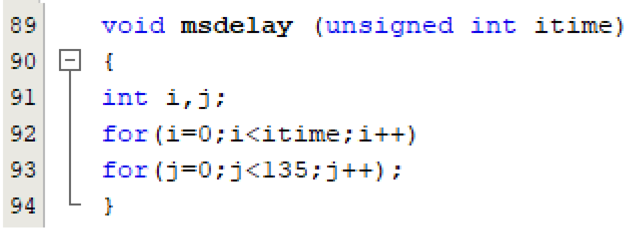
```c
// Arjunsingh Gautam (22070123043)
#include <xc.h>
#include <stdio.h>
#pragma config OSC=HS
#pragma config PWRT=OFF
#pragma config WDT=OFF
#pragma config DEBUG=OFF, LVP=OFF

void lcdcmd(unsigned char value);
void lcddata(unsigned char value);
void msdelay(unsigned int itime);
#define ldata PORTD
#define rs PORTCbits.RC3
#define rw PORTCbits.RC4
#define en PORTCbits.RC5

void main(void)
{
unsigned int i, d;
unsigned char val, temp[3];
unsigned int ADC_Res;
unsigned char ADC_Str[4];
float ADC_Vtg;
unsigned char ADC_Str1[6];
TRISD=0;
PORTD=0;
TRISC=0;
PORTC=0;
```



```c
// ADC Init
ADCON2 = 0b10010101 ; //right justfied, fosc/16, 4 tad
ADCON1 = 0b00001101 ; //an0 & an1 is configured
ADCON0bits.ADON = 1;  // turn on the adc
msdelay(15);
lcdcmd(0x38);
msdelay(15);
lcdcmd(0x0E);
msdelay(15);
lcdcmd(0x01);
msdelay(15);
lcdcmd(0x06);
msdelay(15);

while(1)
{
// Select the channel
ADCON0bits.CHS = 0b000;
// start the adc
ADCON0bits.GO = 1;
while (ADCON0bits.DONE ==1);
ADC_Res = (unsigned int) ADRESH << 8;
ADC_Res |= (unsigned int) ADRESL;
sprintf (ADC_Str, "%d", ADC_Res);
lcdcmd (0x81);
lcddata(ADC_Str[0]);
lcddata(ADC_Str[1]);
lcddata(ADC_Str[2]);
lcddata(ADC_Str[3]);
ADC_Vtg = (float) ADC_Res * (5.0 / 1023.0);
```



```c
ADC_Vtg = (float) ADC_Res * (5.0 / 1023.0);
sprintf (ADC_Str1, "%0.3f", ADC_Vtg);
lcdcmd (0xC1);
lcddata(ADC_Str1[0]);
lcddata(ADC_Str1[1]);
lcddata(ADC_Str1[2]);
lcddata(ADC_Str1[3]);
lcddata(ADC_Str1[4]);
lcddata(ADC_Str1[5]);
msdelay(250);
}
}
void lcdcmd (unsigned char value)
{
ldata=value;
rs=0;
rw=0;
en=1;
msdelay(10);
en=0;
}
void lcddata (unsigned char value)
{
ldata=value;
rs=1;
rw=0;
en=1;
msdelay(10);
en=0;
}
```



```c
void msdelay (unsigned int itime)
{
int i,j;
for(i=0;i<itime;i++)
for(j=0;j<135;j++);
}
```