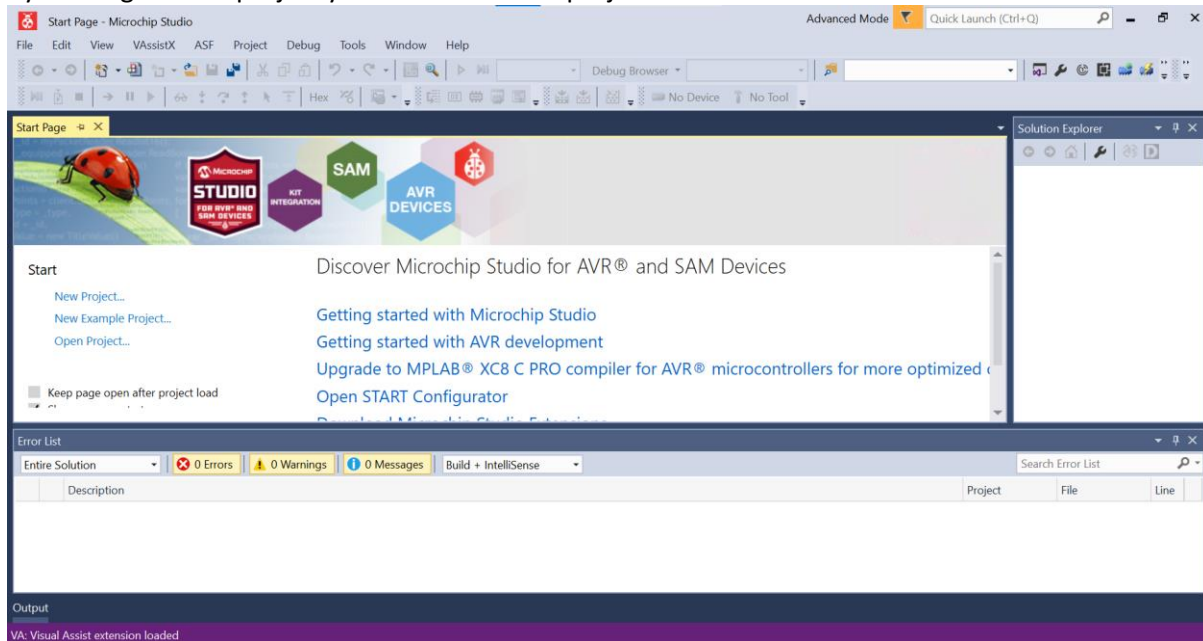


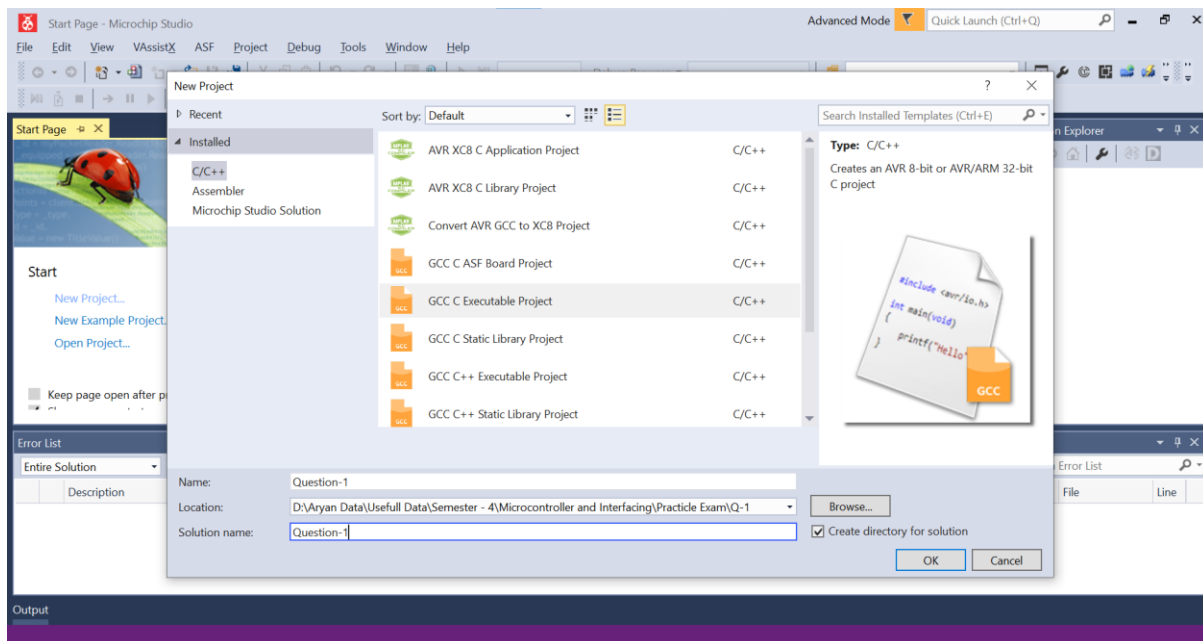
Name :- Aryan Langhanoja
Enrollment No :- 92200133030

Q-1 :

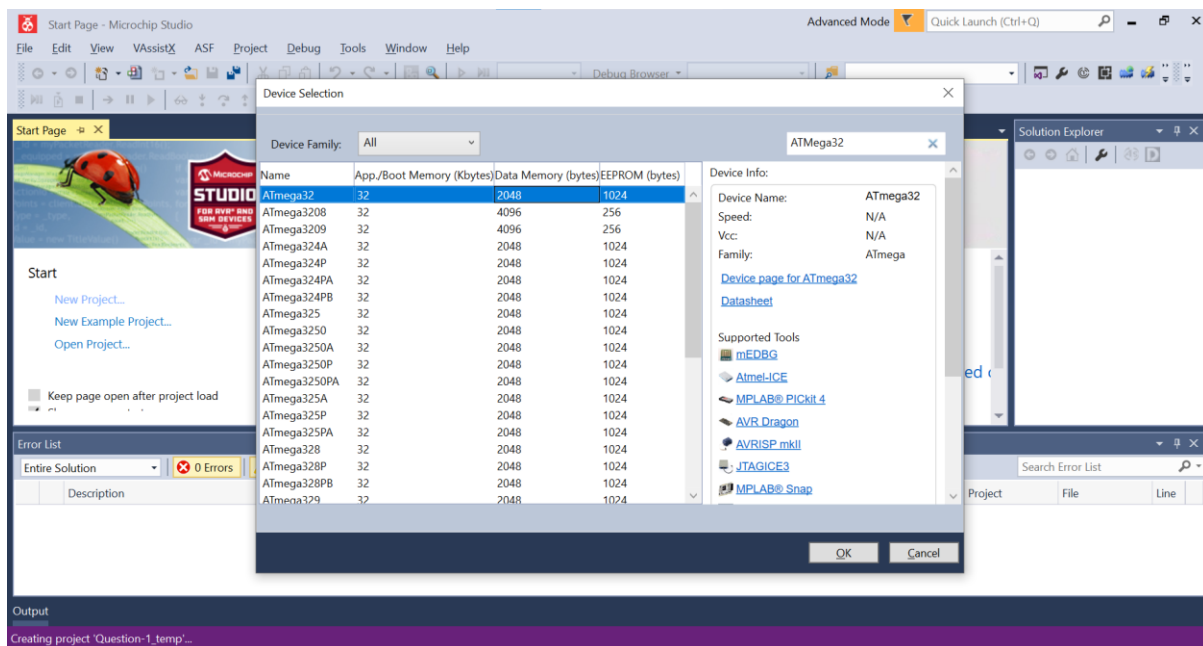
By Clicking on new project you can create new project



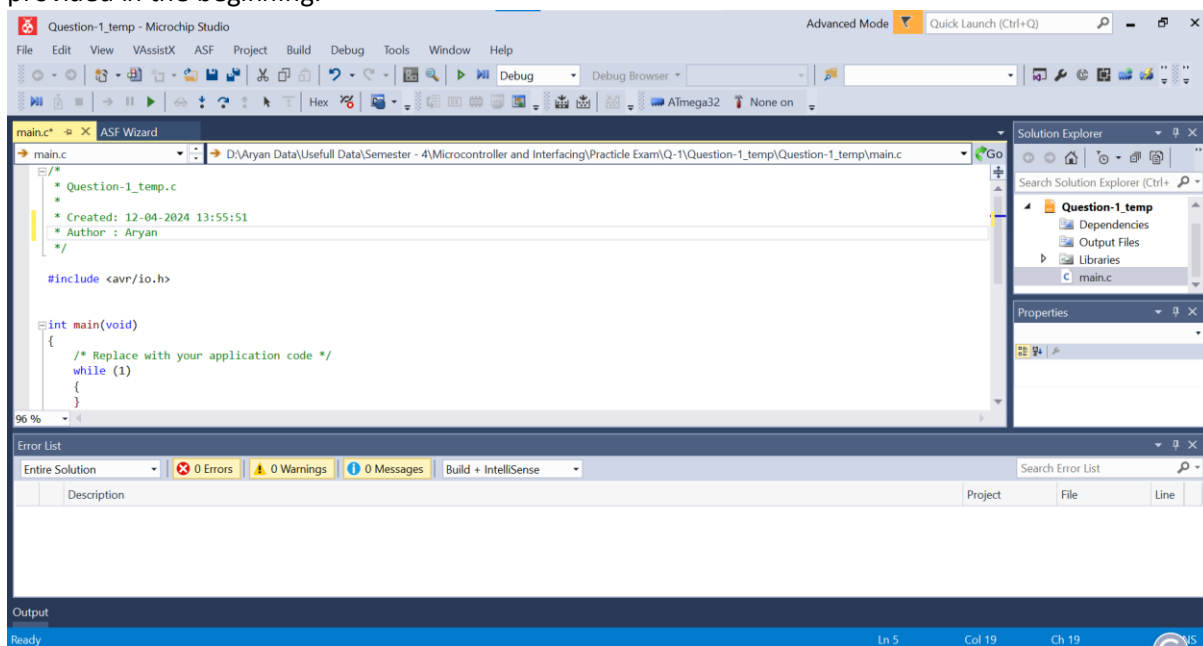
Here enter your project name and path select GCC C EXECUTABLE OPTION and Hit “OK”.



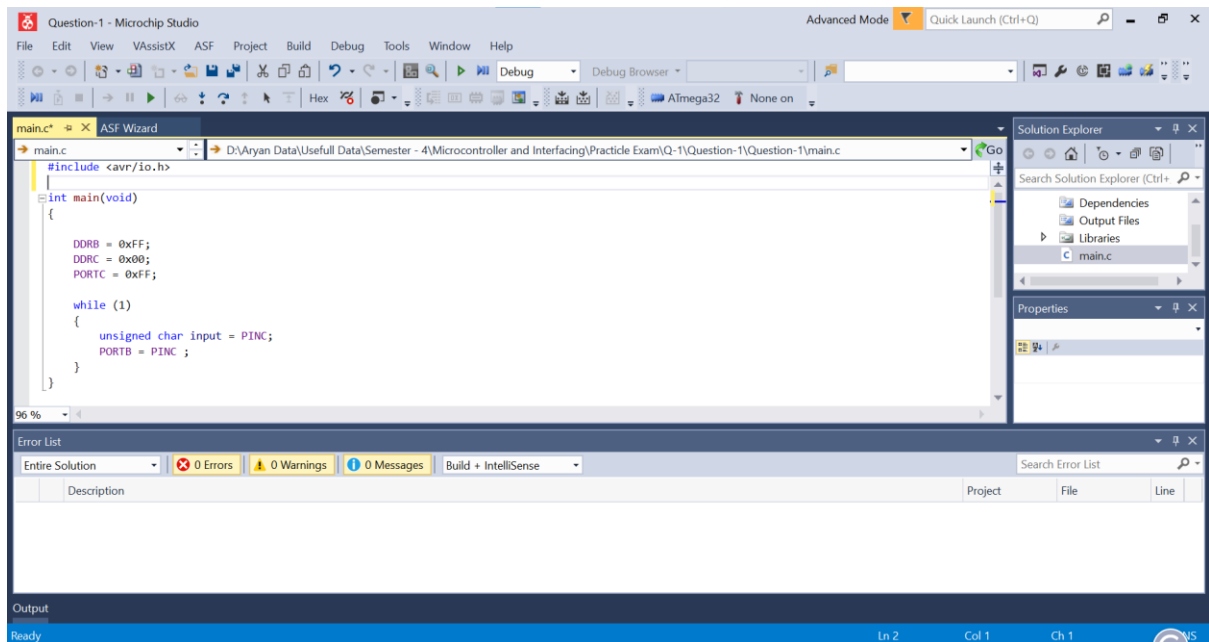
From Here Select Development Board ATmega32 and Hit “OK”



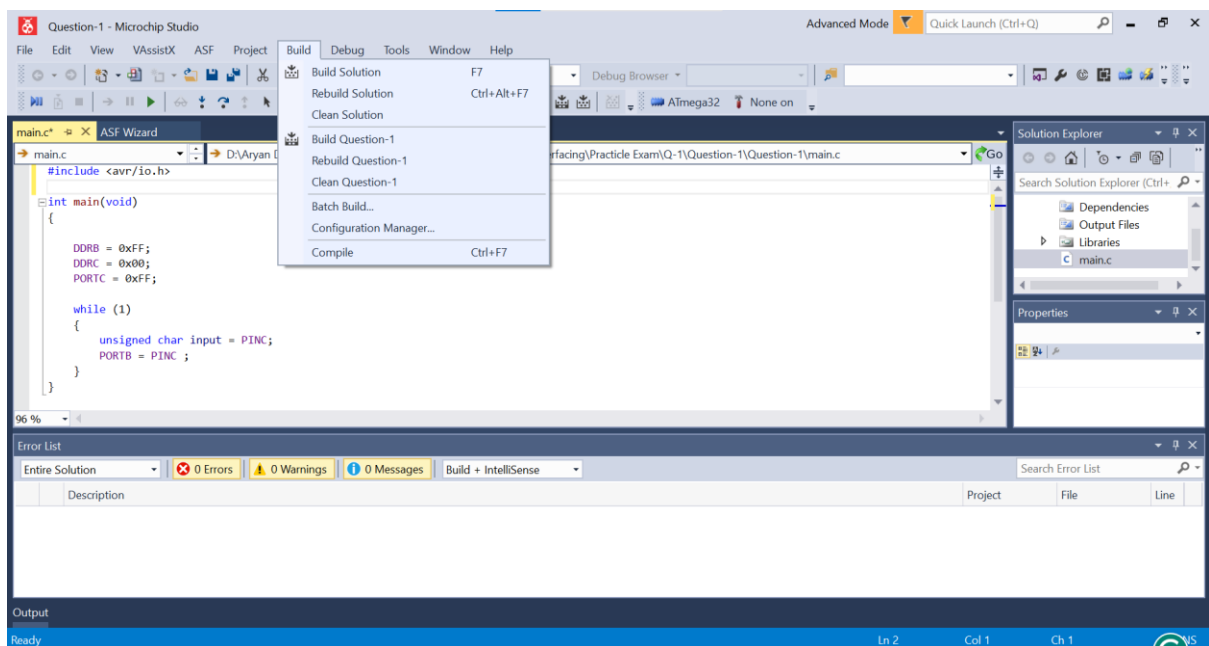
You Will See this type of interface by hitting ok it means your project is created at the path you provided in the beginning.



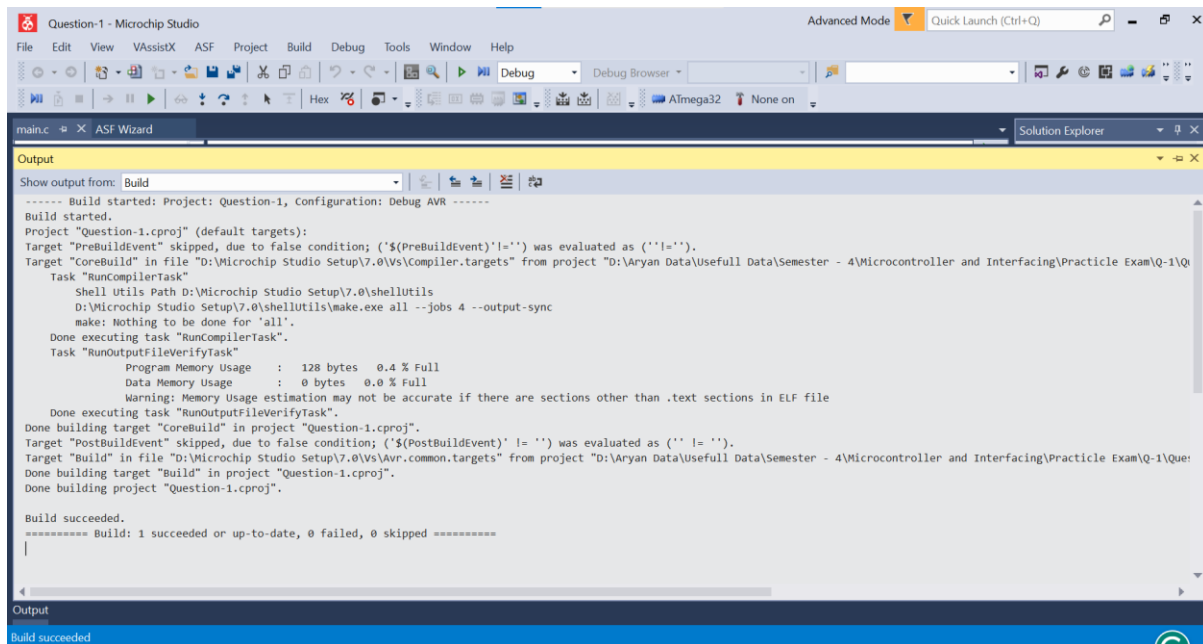
Here you Have to write a program.



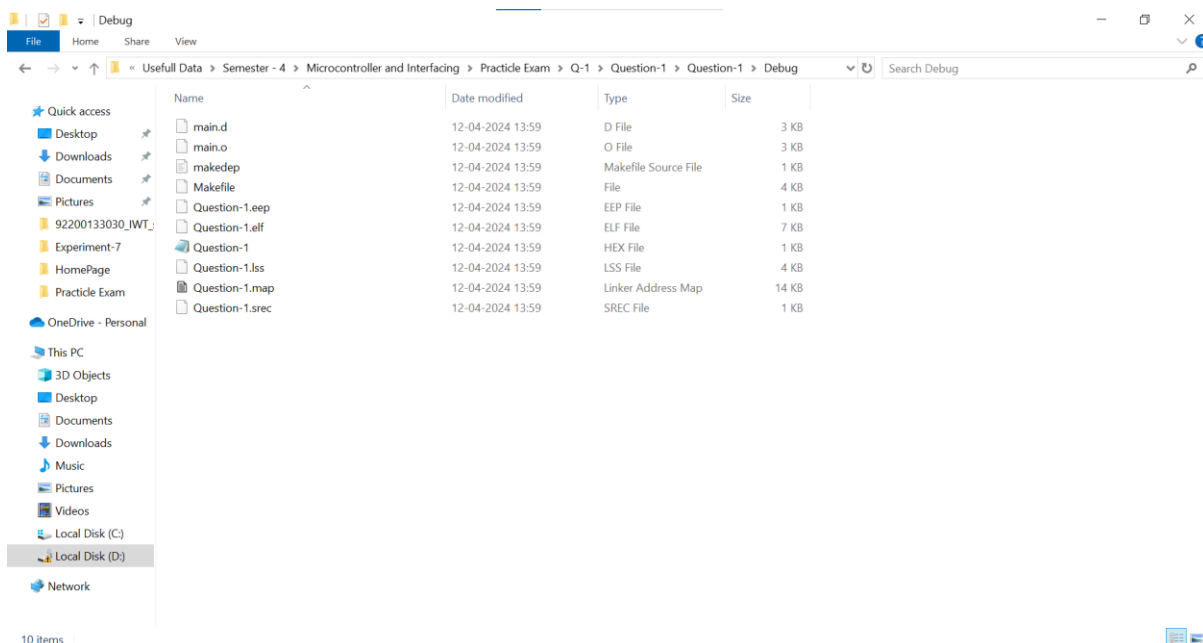
After programm is written click on Build button



Click on Build Solution after successful of building project you will see this type of text in terminal :



Now go to the project location/project_name/debug/project_name.hex
This hex file is generated you have to flash it in your controller board



Q-2 :-

$N = \text{needed Delay} / 0.125\mu\text{s}$ (time period of one cycle)

$\text{TCNT} = 255 - N + 1$

$37(\text{Decima of } 0x25) = 256 - \text{Delay}/0.125$

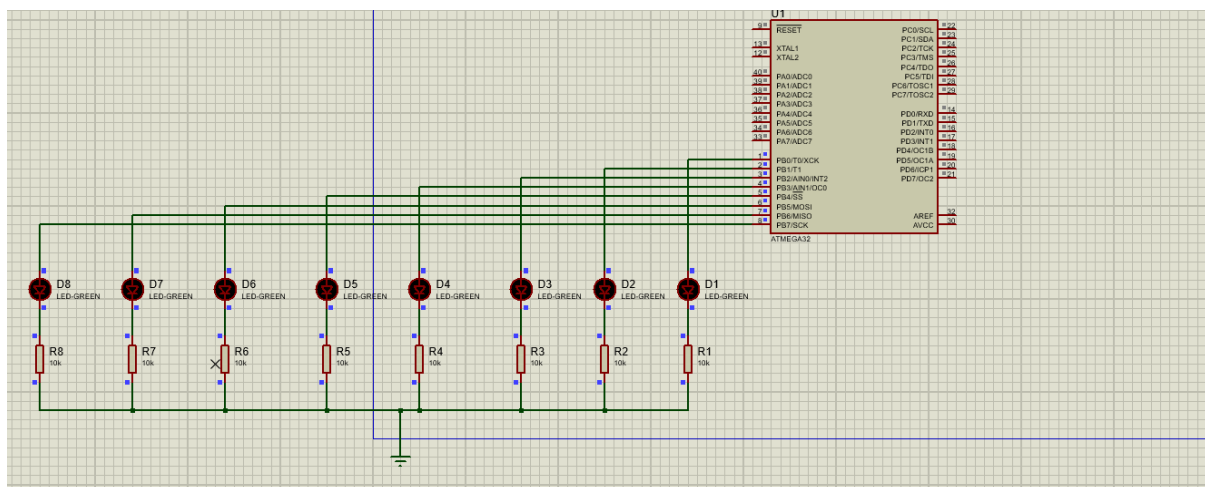
$\text{Delay} = 27.375\mu\text{s}$

To use prescaler of 1024 we should load TCCR0 with 0x05.

Updated Code :

```
void T0delay()
{
    TCNT0 = 0x25;
    TCCR0 = 0x05; // Updated from 0x01 to 0x05
    while((TIFR&0x01)==0);
    TCCR0 = 0;
    TIFR = 0x1;
}
```

Q-3



Programm :-

```
/*
 * Question-3.c
 *
 * Created: 12-04-2024 14:09:40
 * Author : Aryan
 */

#include <avr/io.h>
#include <util/delay.h>
#include <avr/delay.h>
#define F_CPU 16000000UL
```

```

int main(void)
{
    DDRB = 0xFF;
    for(int i = 0 ; i < 5 ; i++) {
        PORTB = 0xFF;
        _delay_ms(1000);
        PORTB = 0x00;
        _delay_ms(1000);
    }

    for(int i = 0 ; i < 5 ; i++) {
        PORTB = 0xF0 ;
        _delay_ms(1000);
        PORTB = 0x0F;
        _delay_ms(1000);
    }

    for(int i = 0 ; i < 5 ; i++) {
        PORTB = 0xCC ;
        _delay_ms(1000);
        PORTB = 0x33;
        _delay_ms(1000);
    }

    for(int i = 0 ; i < 5 ; i++) {
        PORTB = 0xAA ;
        _delay_ms(1000);
        PORTB = 0x55;
        _delay_ms(1000);
    }

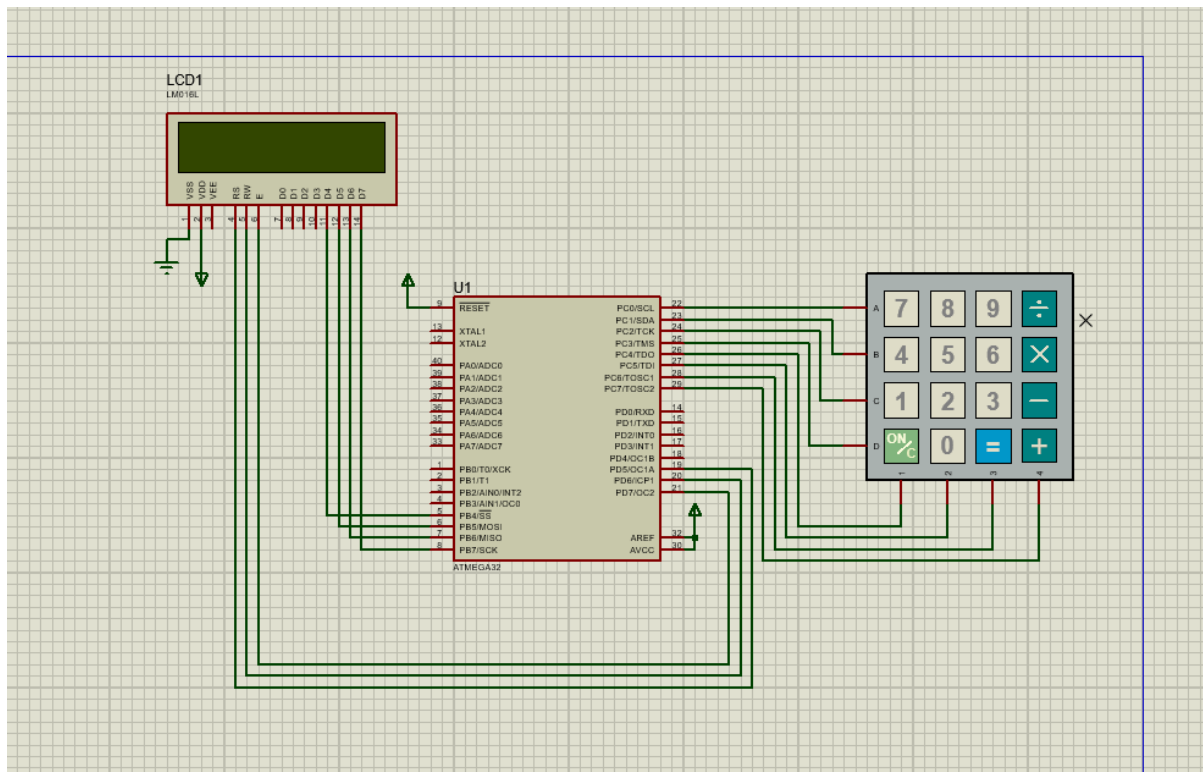
    PORTB = 0x80;
    for(int i = 0 ; i < 8 ; i++) {
        PORTB = 1<<i;
        _delay_ms(1000);
    }

    while(1);

    return 0;
}

```

Q-4 :-



Programm :-

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#define LCD PORTB
#define EN 7
#define RW 6
#define RS 5
unsigned char keypad();
void lcdcmd(unsigned char cmd)
{
    PORTD &= ~(1 << RS);
    PORTD &= ~(1 << RW);
    LCD = cmd & 0xF0;
    PORTD |= (1 << EN);
    _delay_ms(1);
    PORTD &= ~(1 << EN);
    LCD = cmd << 4;
    PORTD |= (1 << EN);
    _delay_ms(1);
    PORTD &= ~(1 << EN);
}
void lcddata(unsigned char data)
{
    PORTD |= (1 << RS);
    PORTD &= ~(1 << RW);
    LCD = data & 0xF0;
    PORTD |= (1 << EN);
    _delay_ms(1);
    PORTD &= ~(1 << EN);
    LCD = data << 4;
    PORTD |= (1 << EN);
    _delay_ms(1);
    PORTD &= ~(1 << EN);
}
```

```

}
void lcd_init(){
    DDRA = 0xFF;
    DDRD = 0xFF;
    PORTD &= ~(1 << EN);
    lcdcmd(0x33);
    lcdcmd(0x32);
    lcdcmd(0x28);
    lcdcmd(0x0E);
    lcdcmd(0x01);
    _delay_ms(2);
}

unsigned char keypad()
{
    PORTC = 0b11111110;
    if ((PINC & (1 << PINC4)) == 0)
    {
        _delay_ms(125);
        return '7';
    }
    else if ((PINC & (1 << PINC5)) == 0)
    {
        _delay_ms(125);
        return '8';
    }
    else if ((PINC & (1 << PINC6)) == 0)
    {
        _delay_ms(125);
        return '9';
    }
    else if ((PINC & (1 << PINC7)) == 0)
    {
        _delay_ms(125);
        return '/';
    }
    PORTC = 0b11111101;
    if ((PINC & (1 << PINC4)) == 0)
    {
        _delay_ms(125);
        return '4';
    }
    else if ((PINC & (1 << PINC5)) == 0)
    {
        _delay_ms(125);
        return '5';
    }
    else if ((PINC & (1 << PINC6)) == 0)
    {
        _delay_ms(125);
        return '6';
    }
    else if ((PINC & (1 << PINC7)) == 0)
    {
        _delay_ms(125);
        return '*';
    }
    PORTC = 0b11111101;
    if ((PINC & (1 << PINC4)) == 0)
    {
        _delay_ms(125);
        return '1';
    }
}

```



```

}
else if ((PINC & (1 << PINC5)) == 0)
{
    _delay_ms(125);
    return '2';
}
else if ((PINC & (1 << PINC6)) == 0)
{
    _delay_ms(125);
    return '3';
}
else if ((PINC & (1 << PINC7)) == 0)
{
    _delay_ms(175);
    return '-';
}
PORTC = 0b11110111;
if ((PINC & (1 << PINC4)) == 0)
{
    _delay_ms(125);
    return 'C';
}
else if ((PINC & (1 << PINC5)) == 0)
{
    _delay_ms(125);
    return '0';
}
else if ((PINC & (1 << PINC6)) == 0)
{
    _delay_ms(125);
    return '=';
}
else if ((PINC & (1 << PINC7)) == 0)
{
    _delay_ms(125);
    return '+';
}
return 0;
}

int main(void)
{
    unsigned char x;
    DDRC = 0x0F;
    _delay_ms(1);
    PORTC = 0xF0;
    lcd_init();
    PORTC = 0xF0;
    _delay_ms(25);
    if (PINC != 0xF0)
    {
        char password[4] = "123x";
        int correct = 0 ;

        for(int i = 0 ; i < 4 ; i++) {
            x = keypad();
            lcddata(x);

            if(x == password[i]) {
                correct++;
            }
        }
    }
}

```

```

        if(correct == 4) {
            char output[] = "Correct Password";

            for(int i = 0 ; i < sizeof(output) / sizeof(output[0]) ;
i++) {
                lcddata(output[i]);
            }
        }
        else {
            char output[] = "Incorrect Password";

            for(int i = 0 ; i < sizeof(output) / sizeof(output[0]) ;
i++) {
                lcddata(output[i]);
            }
        }
        return 0;
    }
}

```