TECHVOLT SOFTWARE PVT.LTD
Simple But Marvelous...

Embedded System Internship Project

Prepared by: Mahesh Mali

Guided by: Jinukrishnan Techvolt

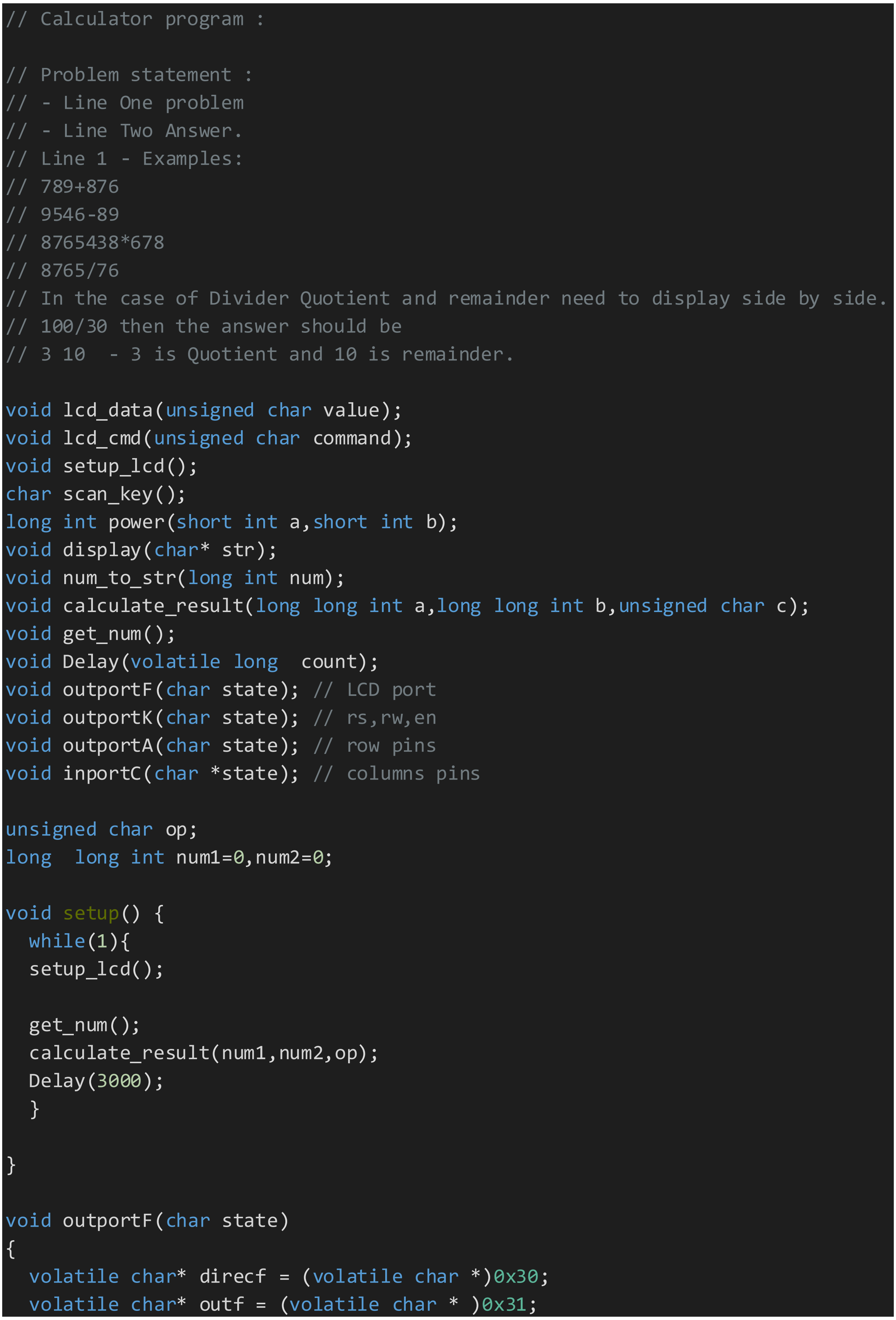
****

Que1) Write the Embedded Code and simulation for making 8x8 digit LCD Calculator with 8051 / ATmega 2560 MCU.

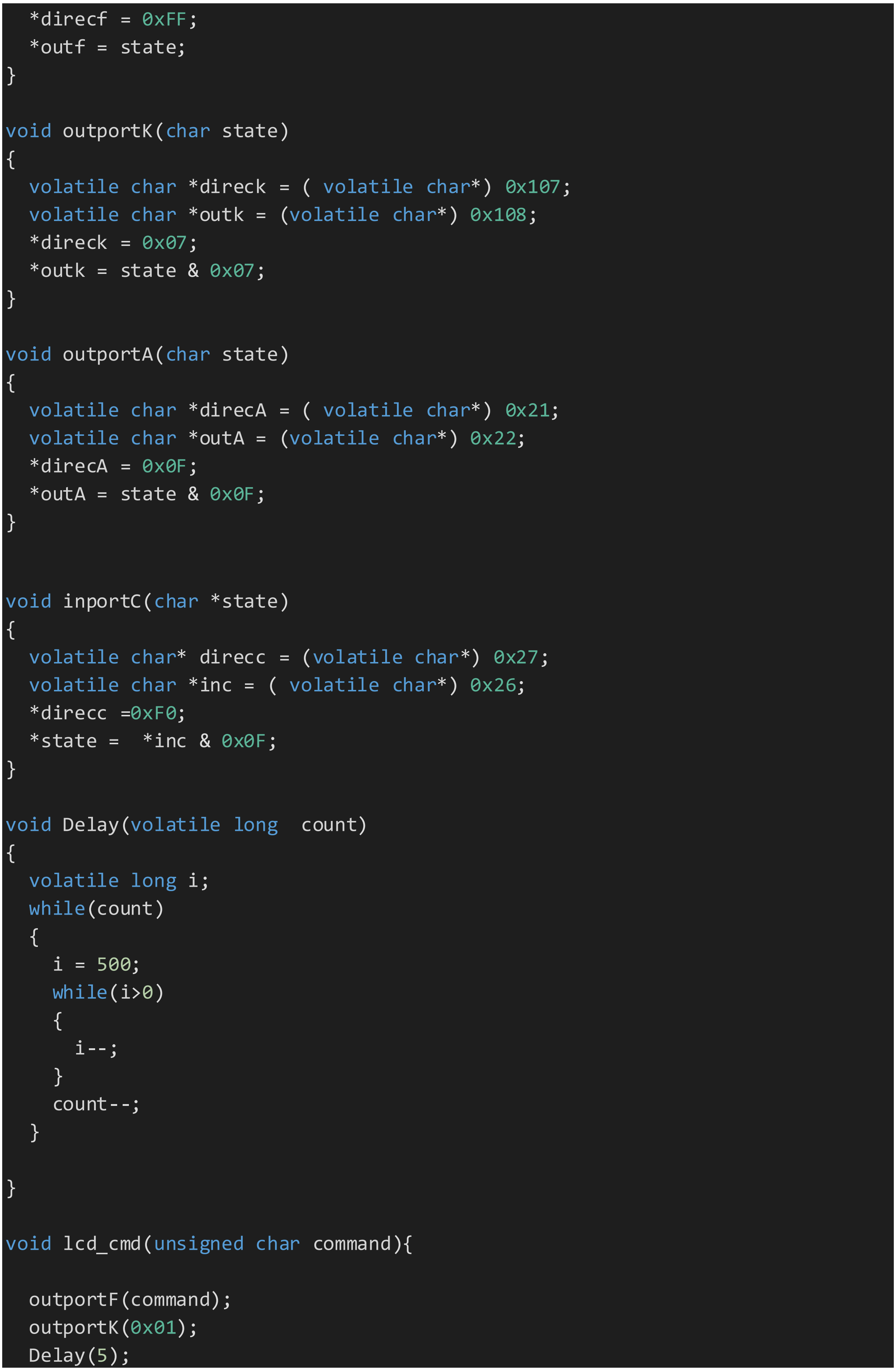
Procedure :

1) In this project I have used ATmega 2560 MCU, LCD display and animated calculator keypad.
2) Here I have driver codes for LCD and keypad differently.
3) LCD driver code include preparing LCD, then sending commands to lcd and then sending data to lcd.
4) Similarly for keypad driver code I have written the scan_key() function. Which in scans the rows and columns to find correct key pressed.
5) Then I have taken first number from user then operator and at last second number .
6) Then I converted those characters to numbers and performed the required operations.
7) At last I reconverted the result to string and displayed it on lcd.

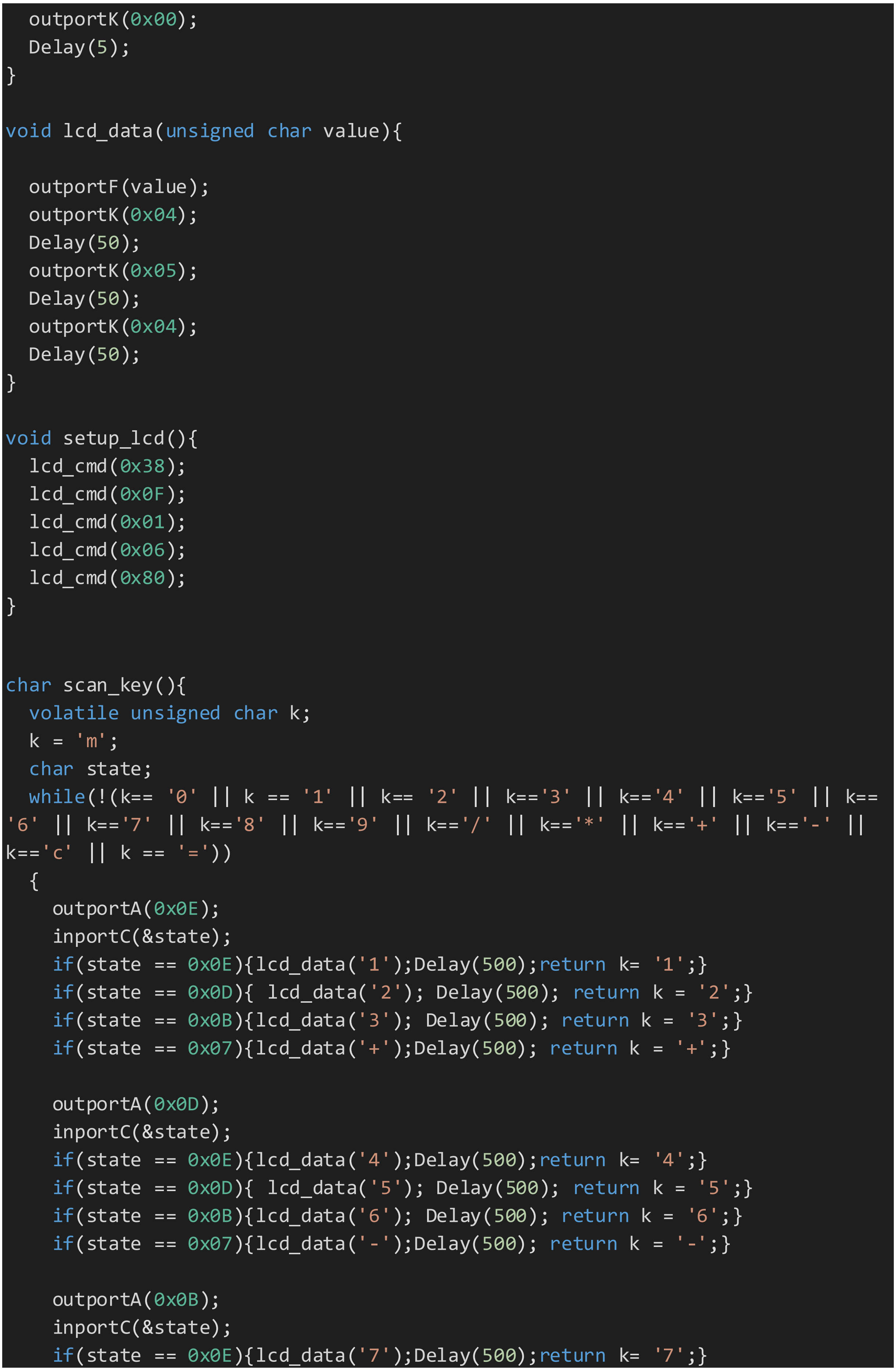
Source code:



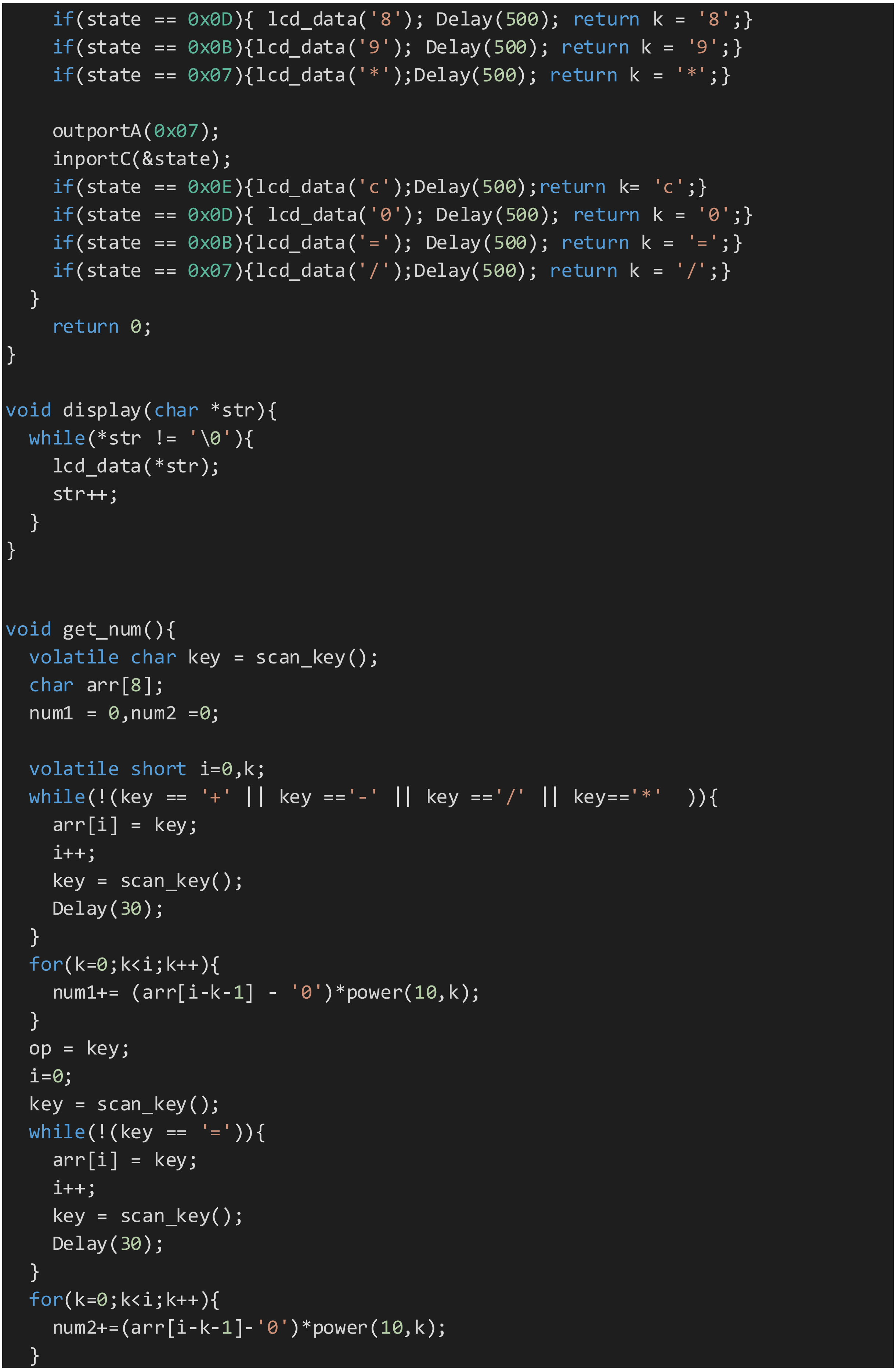
```
// Calculator program :

// Problem statement :
// - Line One problem
// - Line Two Answer.
// Line 1 - Examples:
// 789+876
// 9546-89
// 8765438*678
// 8765/76
// In the case of Divider Quotient and remainder need to display side by side.
// 100/30 then the answer should be
// 3 10  - 3 is Quotient and 10 is remainder.

void lcd_data(unsigned char value);
void lcd_cmd(unsigned char command);
void setup_lcd();
char scan_key();
long int power(short int a,short int b);
void display(char* str);
void num_to_str(long int num);
void calculate_result(long long int a,long long int b,unsigned char c);
void get_num();
void Delay(volatile long  count);
void outportF(char state); // LCD port
void outportK(char state); // rs,rw,en
void outportA(char state); // row pins
void inportC(char *state); // columns pins

unsigned char op;
long  long int num1=0,num2=0;

void setup() {
  while(1){
  setup_lcd();

  get_num();
  calculate_result(num1,num2,op);
  Delay(3000);
  }

}

void outportF(char state)
{
  volatile char* direcf = (volatile char *)0x30;
  volatile char* outf = (volatile char * )0x31;
```



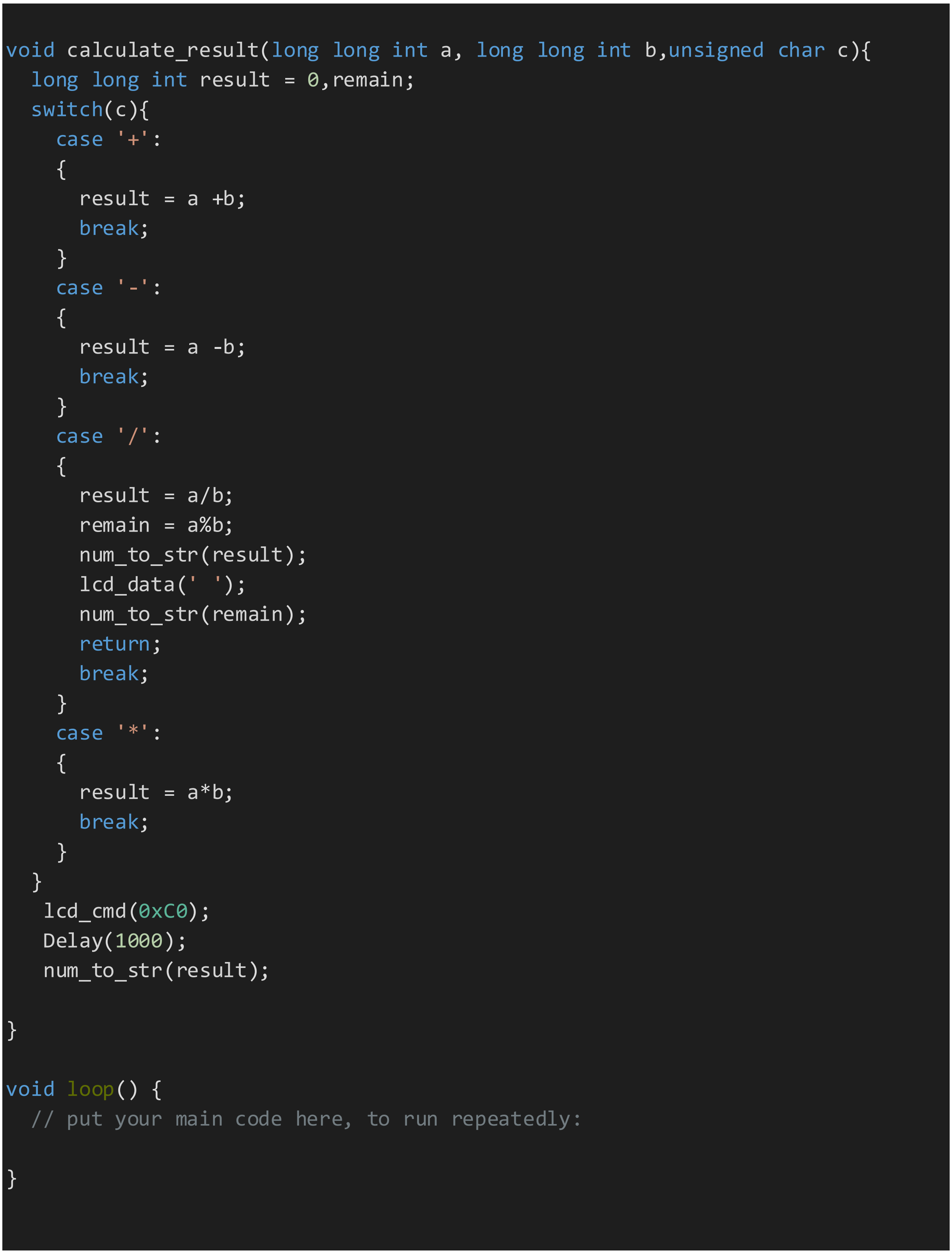
```
  *direcf = 0xFF;
  *outf = state;
}

void outportK(char state)
{
  volatile char *direck = ( volatile char*) 0x107;
  volatile char *outk = (volatile char*) 0x108;
  *direck = 0x07;
  *outk = state & 0x07;
}

void outportA(char state)
{
  volatile char *direcA = ( volatile char*) 0x21;
  volatile char *outA = (volatile char*) 0x22;
  *direcA = 0x0F;
  *outA = state & 0x0F;
}


void inportC(char *state)
{
  volatile char* direcc = (volatile char*) 0x27;
  volatile char *inc = ( volatile char*) 0x26;
  *direcc =0xF0;
  *state =  *inc & 0x0F;
}

void Delay(volatile long  count)
{
  volatile long i;
  while(count)
  {
    i = 500;
    while(i>0)
    {
      i--;
    }
    count--;
  }

}

void lcd_cmd(unsigned char command){

  outportF(command);
  outportK(0x01);
  Delay(5);
```



```
  outportK(0x00);
  Delay(5);
}

void lcd_data(unsigned char value){

  outportF(value);
  outportK(0x04);
  Delay(50);
  outportK(0x05);
  Delay(50);
  outportK(0x04);
  Delay(50);
}

void setup_lcd(){
  lcd_cmd(0x38);
  lcd_cmd(0x0F);
  lcd_cmd(0x01);
  lcd_cmd(0x06);
  lcd_cmd(0x80);
}


char scan_key(){
  volatile unsigned char k;
  k = 'm';
  char state;
  while(!(k== '0' || k == '1' || k== '2' || k=='3' || k=='4' || k=='5' || k== '6' || k=='7' || k=='8' || k=='9' || k=='/' || k=='*' || k=='+' || k=='-' || k=='c' || k == '='))
  {
    outportA(0x0E);
    inportC(&state);
    if(state == 0x0E){lcd_data('1');Delay(500);return k= '1';}
    if(state == 0x0D){ lcd_data('2'); Delay(500); return k = '2';}
    if(state == 0x0B){lcd_data('3'); Delay(500); return k = '3';}
    if(state == 0x07){lcd_data('+');Delay(500); return k = '+';}

    outportA(0x0D);
    inportC(&state);
    if(state == 0x0E){lcd_data('4');Delay(500);return k= '4';}
    if(state == 0x0D){ lcd_data('5'); Delay(500); return k = '5';}
    if(state == 0x0B){lcd_data('6'); Delay(500); return k = '6';}
    if(state == 0x07){lcd_data('-');Delay(500); return k = '-';}

    outportA(0x0B);
    inportC(&state);
    if(state == 0x0E){lcd_data('7');Delay(500);return k= '7';}
```

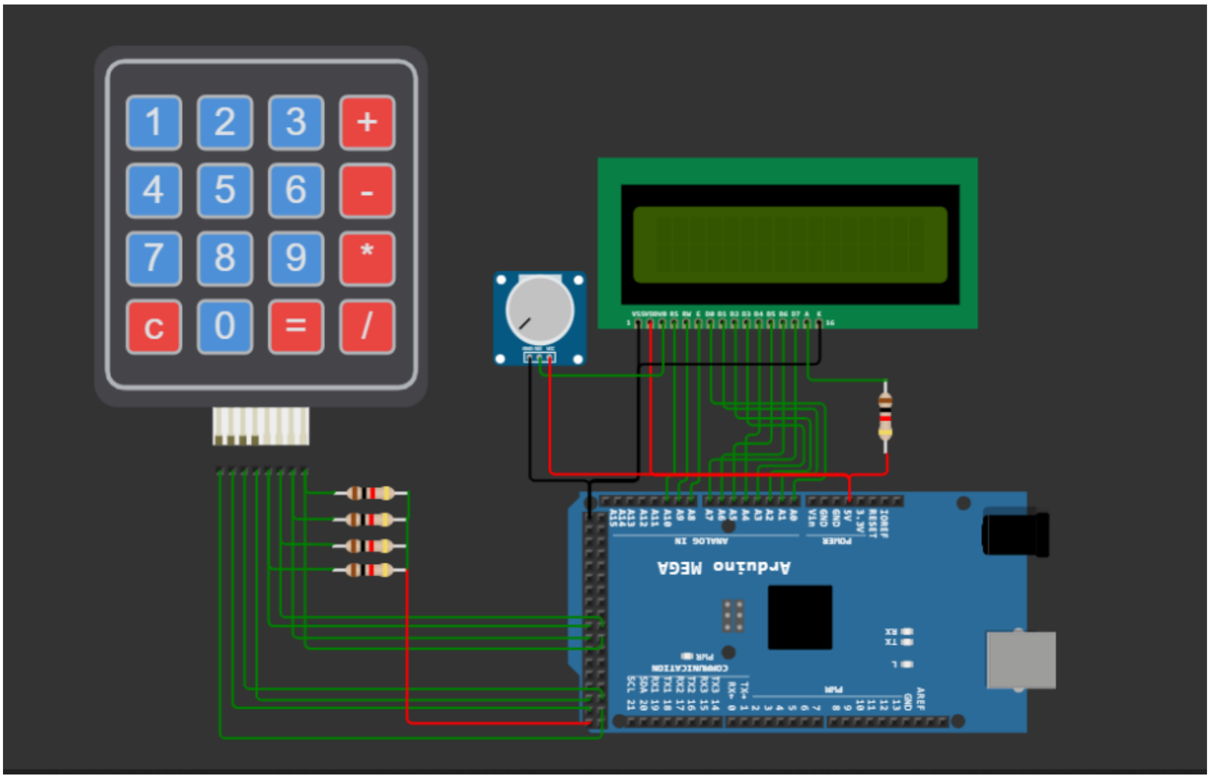


```
    if(state == 0x0D){ lcd_data('8'); Delay(500); return k = '8';}
    if(state == 0x0B){lcd_data('9'); Delay(500); return k = '9';}
    if(state == 0x07){lcd_data('*');Delay(500); return k = '*';}

    outportA(0x07);
    inportC(&state);
    if(state == 0x0E){lcd_data('c');Delay(500);return k= 'c';}
    if(state == 0x0D){ lcd_data('0'); Delay(500); return k = '0';}
    if(state == 0x0B){lcd_data('='); Delay(500); return k = '=';}
    if(state == 0x07){lcd_data('/');Delay(500); return k = '/';}
  }
    return 0;
}

void display(char *str){
  while(*str != '\0'){
    lcd_data(*str);
    str++;
  }
}


void get_num(){
  volatile char key = scan_key();
  char arr[8];
  num1 = 0,num2 =0;

  volatile short i=0,k;
  while(!(key == '+' || key =='-' || key =='/' || key=='*'  )){
    arr[i] = key;
    i++;
    key = scan_key();
    Delay(30);
  }
  for(k=0;k<i;k++){
    num1+= (arr[i-k-1] - '0')*power(10,k);
  }
  op = key;
  i=0;
  key = scan_key();
  while(!(key == '=')){
    arr[i] = key;
    i++;
    key = scan_key();
    Delay(30);
  }
  for(k=0;k<i;k++){
    num2+=(arr[i-k-1]-'0')*power(10,k);
  }
  lcd_cmd(0x01);
  lcd_cmd(0xC0);

}


long int power(short int a,short int b){
  long int result = 1;
  short int count;
  for(count=0;count<b;count++){
    result *= a;
  }
  return result;
}


void num_to_str(long long int real_num){
  long long int num;
  char temp[17];
  char i=0,k;
  char var;

  if(real_num>=0){
    num = real_num;
  }
  else{
    num = -1*real_num;
    lcd_data('-');
  }

  while(num>0){
      temp[i] = ((num%10) + '0');
      i++ ;
      num = num/10;
  }
  if(i==0){
    lcd_data('0');
    return;
  }
  temp[i] = '\0';
  for(k=0;k<i/2;k++){
    var = temp[k];
    temp[k] = temp[i-k-1];
    temp[i-k-1] = var;
  }
  display(temp);
}
```



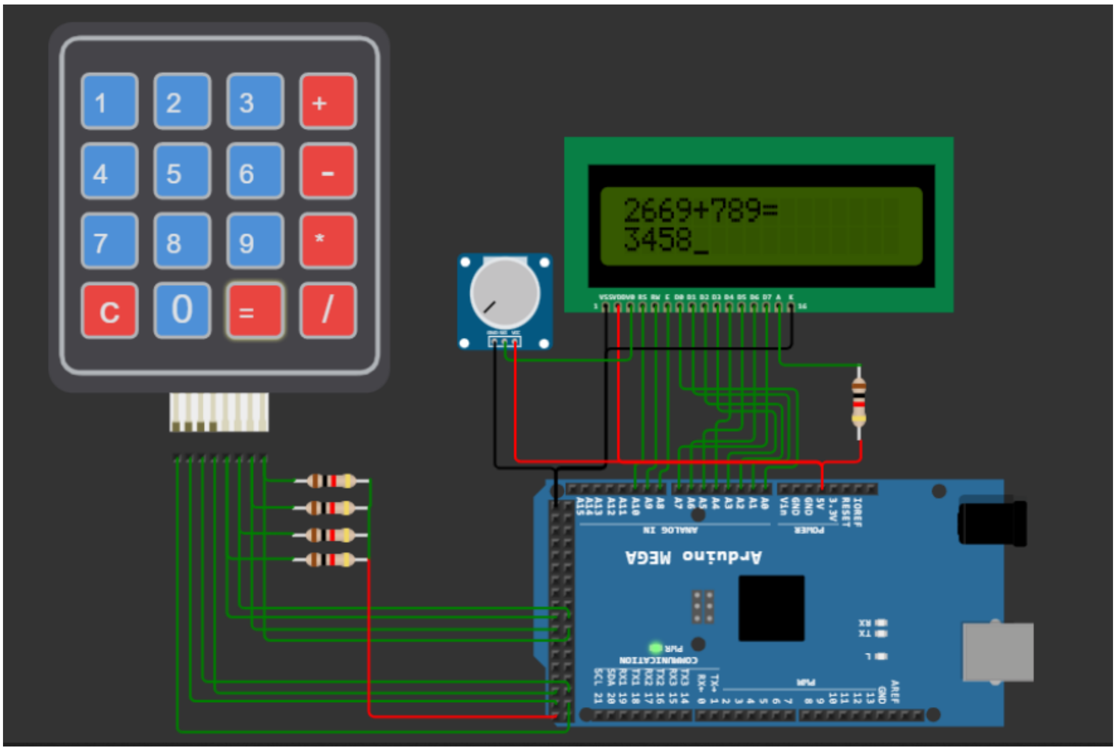
```
void calculate_result(long long int a, long long int b,unsigned char c){
  long long int result = 0,remain;
  switch(c){
    case '+':
    {
      result = a +b;
      break;
    }
    case '-':
    {
      result = a -b;
      break;
    }
    case '/':
    {
      result = a/b;
      remain = a%b;
      num_to_str(result);
      lcd_data(' ');
      num_to_str(remain);
      return;
      break;
    }
    case '*':
    {
      result = a*b;
      break;
    }
  }
   lcd_cmd(0xC0);
   Delay(1000);
   num_to_str(result);

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Simulation connections:



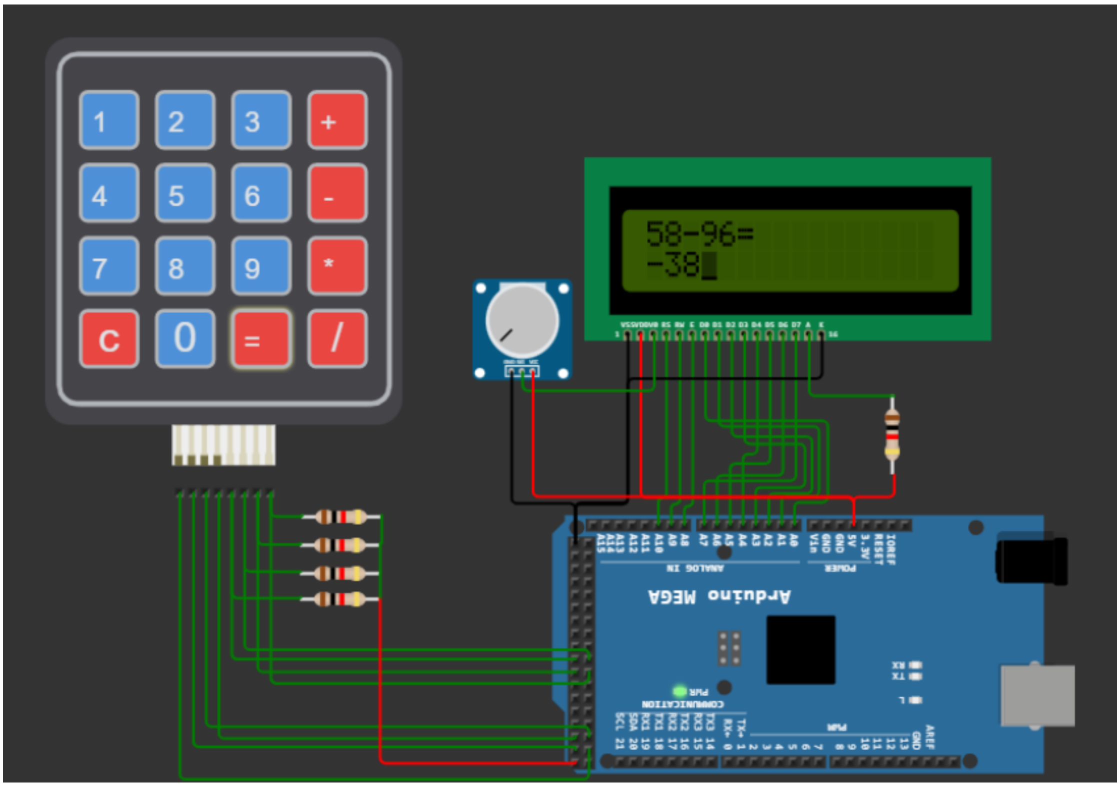
Test Input Results :

For Addition :



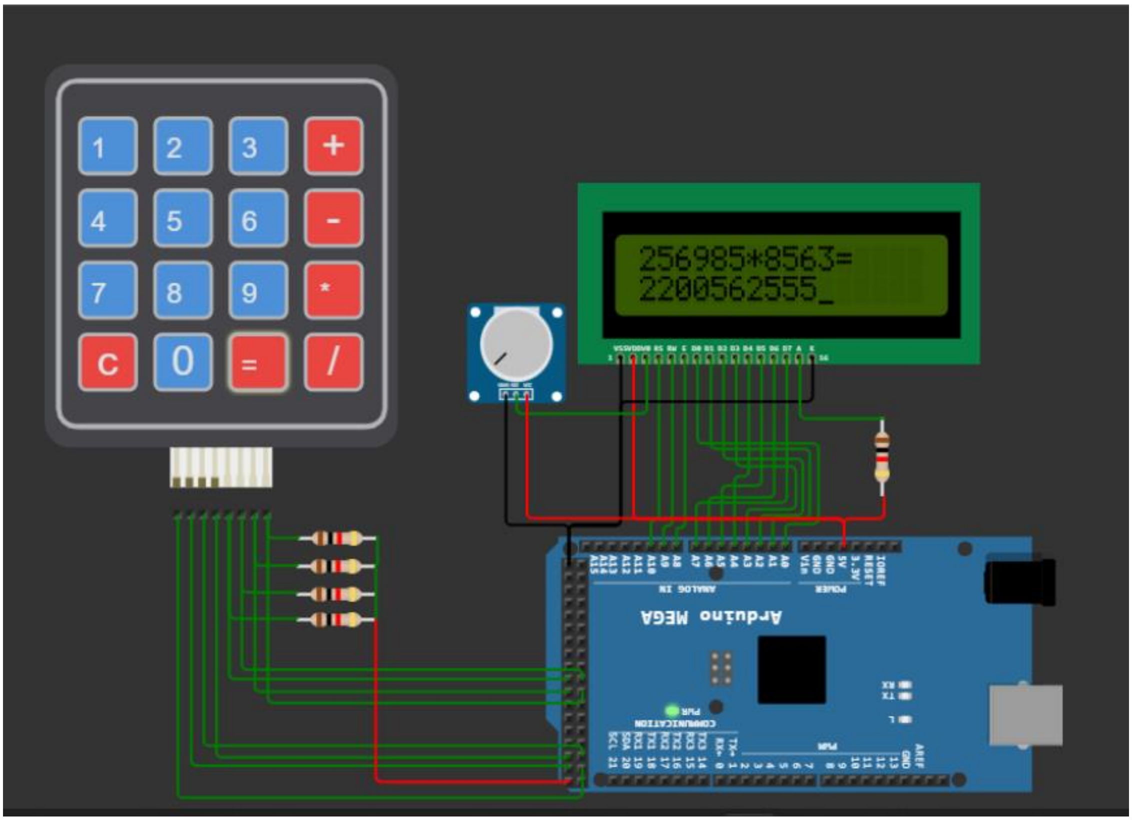
```
2669+789=
3458_
```

For Subtraction :



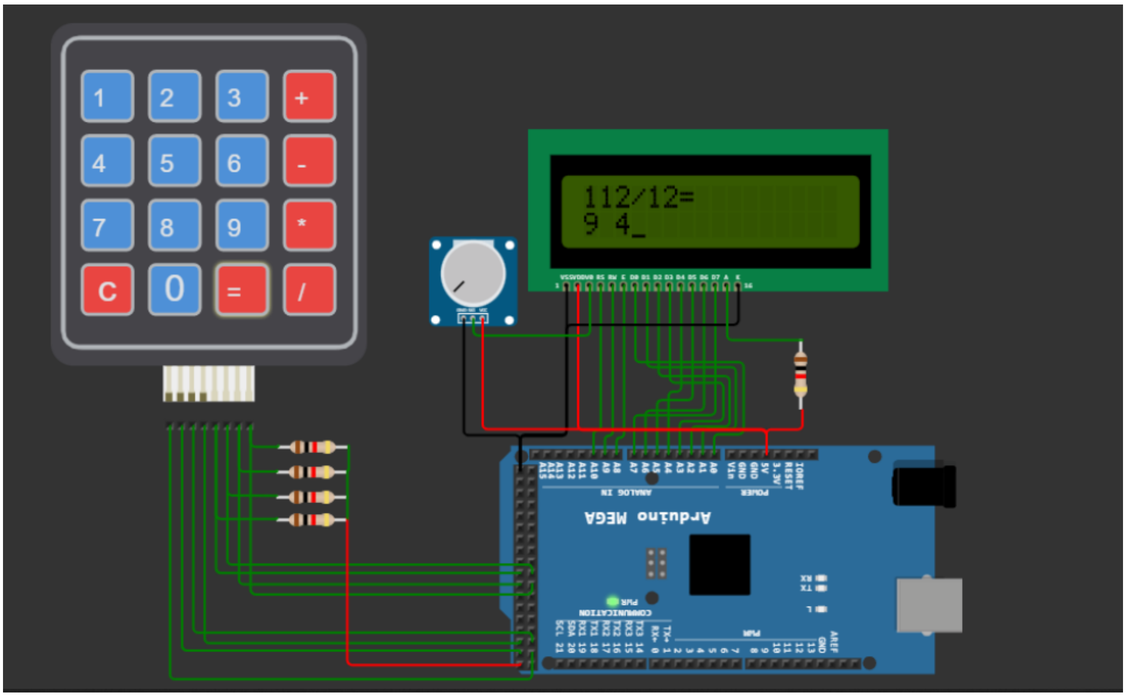
```
58-96=
-38
```

For Multiplication :



```
256985*8563=
2200562555_
```

For Division:



CONCLUSIONS :

1) From this Project I came to know about interfacing LCD display and calculator keypad with ATmega 2560 MCU.
2) Also I learned about how to use I/O port of ATmega 2560 MCU to make useful application.

Link to project simulation platform :
https://wokwi.com/arduino/projects/313411629111837248

********************** Thank you *******************************