MATHEMATICS SMART QUIZ

A Course project report submitted

in partial fulfilment of requirement

of

SMART SYSTEM DESIGN

by

S.AMULYA	**(19K41A04B6)**
D.MEGHANA	**(19K41A04A6)**
V.PARASHURAM	**(19K41A04B8)**
G.GANESH	**(19K41A0499)**

Under the guidance of

Mr.Ch.Rajendra Prasad

Assistant Professor, Department of ECE

SR Engineering College

Innovation . Creativity . Entrepreneurship

ABSTRACT

The mathematics smart quiz project aims to create a smart math quiz to test the mathematical problem solving abilities of students. Through this quiz we aim to challenge and enhance the thinking ability and problem solving capability of the students by asking them questions on fundamental mathematics. In our quiz, the Arduino generates math questions based on the type of operation the student has selected. Whenever the student gives an answer, they can see if their answer is correct or not on the LCD screen. If their answer is correct, one point is awarded and if their answer is wrong, no points are awarded. In addition to that, we also display the correct answer for the student's reference. Through our quiz, students can answer as many problems as they want and can view their score whenever they like.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In the world of education there are many different ways to teach and to consolidate what has been learned. In years gone by, children were expected to memorize dates, formulae and figures by rote. But this method of teaching can be quite dull, to say the least! One tool becoming more common, especially in online education, is quizzes. There are many reasons why quizzes help students to learn. Firstly, Quizzes improve concentration because; when we're playing a quiz we have to keep our mind on what we are doing. Secondly, most children find that quizzes are fun and they also help us to retain information. Reading information as a way of learning does have its uses. But reading information and then taking a quiz is much more effective. Forcing your brain to retrieve data ensures that it becomes 'embedded' for use in the future. Lastly, a quiz helps the students to improve confidence in them.

In our project, we have developed a mathematics quiz using Arduino. We made a smart quiz which is interactive with the user, as the user gets to pick the type of question they want. This type of interactive quiz keeps the interest of the student. Through math smart quiz, we challenge the mathematical abilities of the student and make practice feel like a play.

Advantages:

The advantages of mathematics smart quiz are:

- The quiz is interactive so, it can keep the interest of the user and helps them to indulge in the experience of learning.
- Questions will be asked based on the mathematical operation the student chooses.
- If the user gives incorrect answer, correct answer is displayed for their reference. So that they can learn and not repeat that mistake again.
- As there is no time limit, the student can feel less stressful to answer the question and they can think thoroughly before giving their answer.
- Score can be viewed whenever the user wishes to, and they can start the quiz from the beginning, again.
- The questions are generated automatically, so we cannot predict which question comes next.
- The quiz is color coded, that is, the neo pixel jewel shows green and red colors based on the response of the student.

Disadvantages:

- User can choose the question only from 4 basic mathematical operations.

- The keypad does not have a clear screen button. So, once if the user gives an answer, it cannot be erased.
- Limited number of keys on keypad.

1.2 OVERVIEW OF PROJECT

- The math quiz is smart and interactive. It tests the students' math solving ability.
- Questions are generated based on the type of operation the user chooses.
- Automatic evaluation system. Correct answer is displayed if the user is wrong.
- Score is displayed based on the number of correct answers.

1.3 OBJECTIVES

- The objective of this project is to create a smart mathematics quiz which is both fun and knowledgeable. We aim to test and challenge the problem solving and thinking capability of students.

CHAPTER 2

PROJECT DESCRIPTION

2.1 BLOCK DIAGRAM OF THE PROJECT

As shown in the below Fig.2.1, our project consists of Arduino UNO, keypad, LCD and neo pixel jewel. The input device is keypad. The output devices are LCD and neo pixel jewel. Keypad is used to enter the numbers. LCD is used to display the questions, answers and score. Neo pixel jewel is used to indicate correct answers, wrong answers and score by green, red and yellow light respectively.

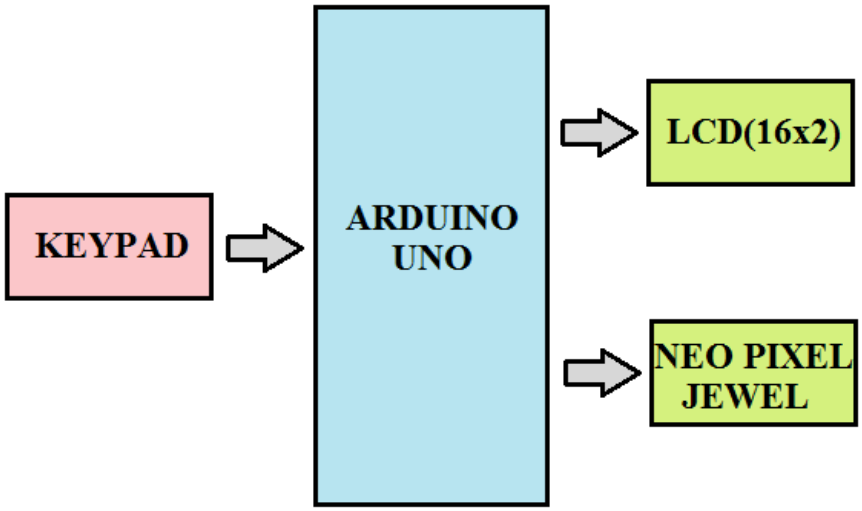

Fig.2.1 Block Diagram

2.2HARDWARE DESCRIPTION

2.2.1Arduino UNO

Description

The Arduino Uno is an open source micro controller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The Uno board is the first in a series of USB-based Arduino boards. The word UNO means one in Italian and was chosen to mark the initial release of Arduino Software. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins, 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB Cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts.

Technical specifications

- Microcontroller: Microchip ATmega328P
- Operating Voltage: 5 Volts
- Input Voltage (recommended): 7 to 12 Volts
- Input Voltage (limits): 6 to 20 Volts
- Digital I/O Pins: 14
- Analog Input Pins: 6
- DC Current per I/O Pin: 20mA
- DC Current 3.3V Pin: 50mA
- Flash memory: 32KB of which 0.5Kb used by bootloader
- SRAM: 2KB
- EEPRON: 1KB
- Length: 68.6 mm
- Width: 53.4 mm
- Weight: 25 g

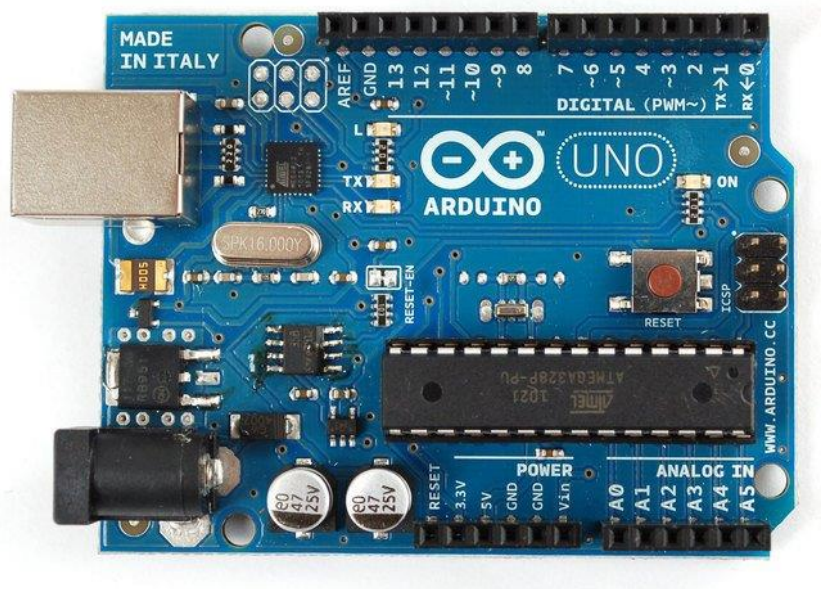

Fig. 2.2 Arduino Uno

Applications:

- C-STEM Studio, a platform for hands-on integrated learning of computing, science, technology, engineering and mathematics (C-STEM) with robotics.
- Arduboy, a handheld game console based on Arduino.

- Low cost data glove for virtual reality applications
- Arduinome, a MIDI controller device that mimics Monome.
- Ardupilot, drone software and hardware.
- ArduSat, a cubesat based on Arduino.
- Water quality testing platform
- Automatic titration system based on Arduino and stepper motor
- Impedance sensor system to detect bovine milk adulteration
- Homemade CNC using Arduino and DC motors with close loop control by Homofaciens
- DC motor control using Arduino and H-Bridge

2.2.2 Keypad

Description

A keypad is a set of buttons arranged in a block or "pad" which bear digits, symbols or alphabetical letters. Pads mostly containing numbers are called a numeric keypad. The 4 x 4 matrix keypad usually is used as input in a project. It has 16 keys in total, which means the same input values. It is ultra-thin, easy to interface with any microcontroller and has an adhesive backing for easy mounting for a variety of applications.

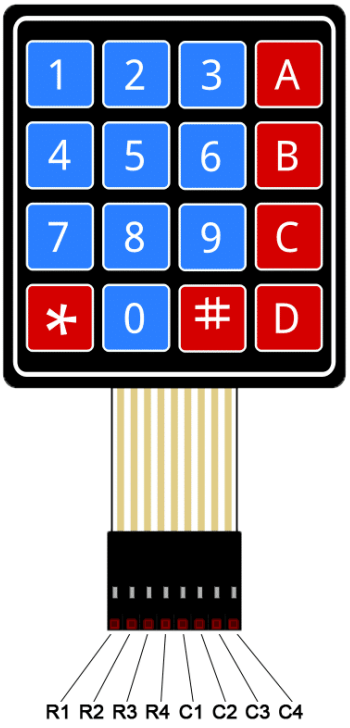

Fig. 2.3 Keypad

Specifications

- Maximum Rating: 24 VDC, 30 mA.
- Interface: 8-pin access to 4x4 matrix.
- Operating temperature: 32 to 122 °F (0 to 50°C)
- Dimensions: Keypad, 2.7 x 3.0 in (6.9 x 7.6 cm)

Applications

- Password protected door security system.
- Wired remote controlled robot.
- Electronic voting machine.
- Generating external interrupt.
- Musical keypad and many more.
- Multiple DIY projects.

2.2.3 Liquid Crystal Display (LCD)

Description

An LCD (Liquid Crystal Display) screen is an electronic display module and has a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. The 16 x 2 display is capable of displaying 224 different characters and symbols. This LCD has two registers, namely, Command and data.

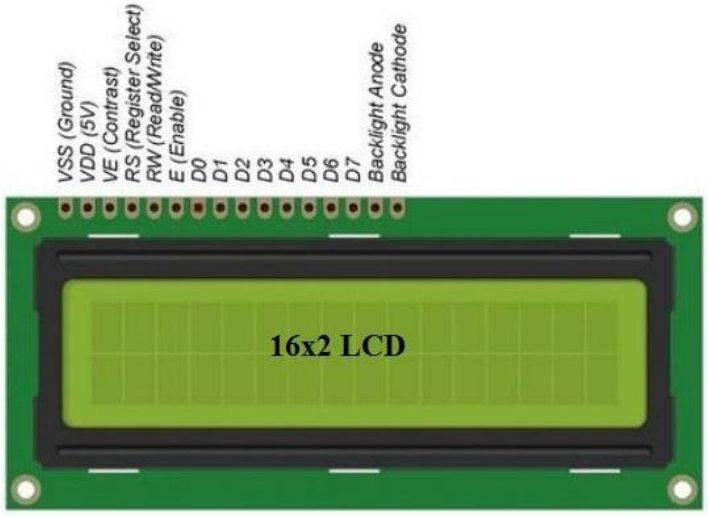

Fig. 2.4 Liquid Crystal Display(LCD)

Specifications

- Operating Voltage is 4.7V to 5.3V.
- Current consumption is 1mA without backlight.
- Alphanumeric LCD display module, meaning can display alphabets and numbers.
- Consists of two rows and each row can print 16 characters..
- Each character is built by a 5×8 pixel box.
- Can work on both 8-bit and 4-bit mode.
- It can also display any custom generated characters.
- Available in Green and Blue Backlight.

2.2.4 Neo pixel Jewel

Description

The NeoPixel LEDs are RGB LED lights with a built driver IC that makes these lights addressable and programmable. The idea was originally coined by Adafruit and since then there are many types of Neo pixels of varying sizes and shapes available in the market. The Neo pixels are small in size with less circuitry and almost no messy wires since the driver IC is embedded into each LED. Each LED has a minimum of RGB light and hence they can be combined to get almost any color of your choice. This makes it a very good choice for wearable electronics and other decorative lights.

Fig. 2.5 Neo pixel jewel

Specifications

- Individually addressable and programmable RGB LEDs
- Flexible and available in different form factors
- Operating voltage: 3.3V to 5V
- Power consumption: 60mA per LED at full brightness
- Communication: PWM through data pin
- Driver IC: WS2812
- Available in many different packages and form factors

Applications

- Wearable electronics
- Linear clock
- Mood Lamp
- Digital Posters
- Electronic/Digital advertisements

2.3 SOFTWARE DESCRIPTION

The software used here is ARDUINO SOFTWARE:

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Writing Sketches:

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB:

Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the ino extension on save.

Verify

Checks your code for errors compiling it.

Upload

Compiles your code and uploads it to the configured board. See uploading below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"

New

Creates a new sketch.

Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.

Save

Saves your sketch.

Serial Monitor

Opens the serial monitor.

Additional commands are found within the five menus: File, Edit, Sketch, Tools,and help.

Programming on Arduino UNO

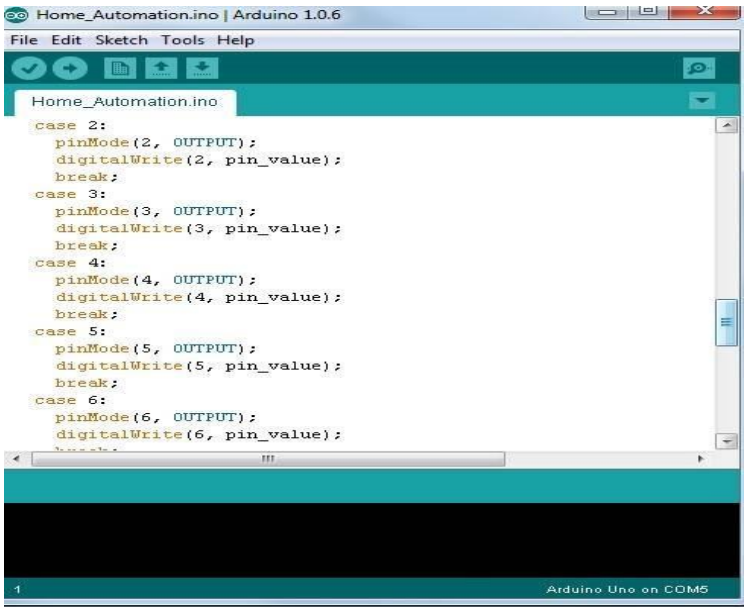

Fig.2.6 Software IDE

In order for the Arduino-Uno board to be able to interact with the application used in this project certain program (code) needs to be uploaded to the Arduino-Uno.

Arduino Company provides user friendly software which allows writing any code for any function wanted to be performed by the Arduino-Uno and upload it to the board. Refer to appendix A for the full source code of the Arduino-Uno board.

CHAPTER 3

CIRCUIT DIAGRAM AND DESCRIPTION

3.1 Working

As shown in the below figures 3.1 and 3.2, the Arduino uno board is connected to a keypad, LCD and neo pixel jewel. When we run the program, the quiz starts. The title is printed on the LCD. Then, we get the options of addition, subtraction, multiplication, division and score menu on the LCD. The user can choose any option from the menu and enter their option through the keyboard. After they choose the operation, a system generated math question is displayed on the LCD. The user can answer the given question by entering their answer through the keypad. If the answer they submitted is correct, a message is displayed on the LCD that they have given the correct answer and neo pixel jewel shows green light. If the answer is wrong, the LCD displays that their answer is wrong, and also shows the correct answer of that question, in addition to that, the neo pixel shows red light. After some delay, the menu holding different options is displayed again on the LCD, from which the user can either choose their next question or they can view their score. When the user choses to see their score, the can see it on the LCD and the neo pixel jewel gives yellow light. A few seconds later, the score is reset to zero and the quiz starts from the beginning again.

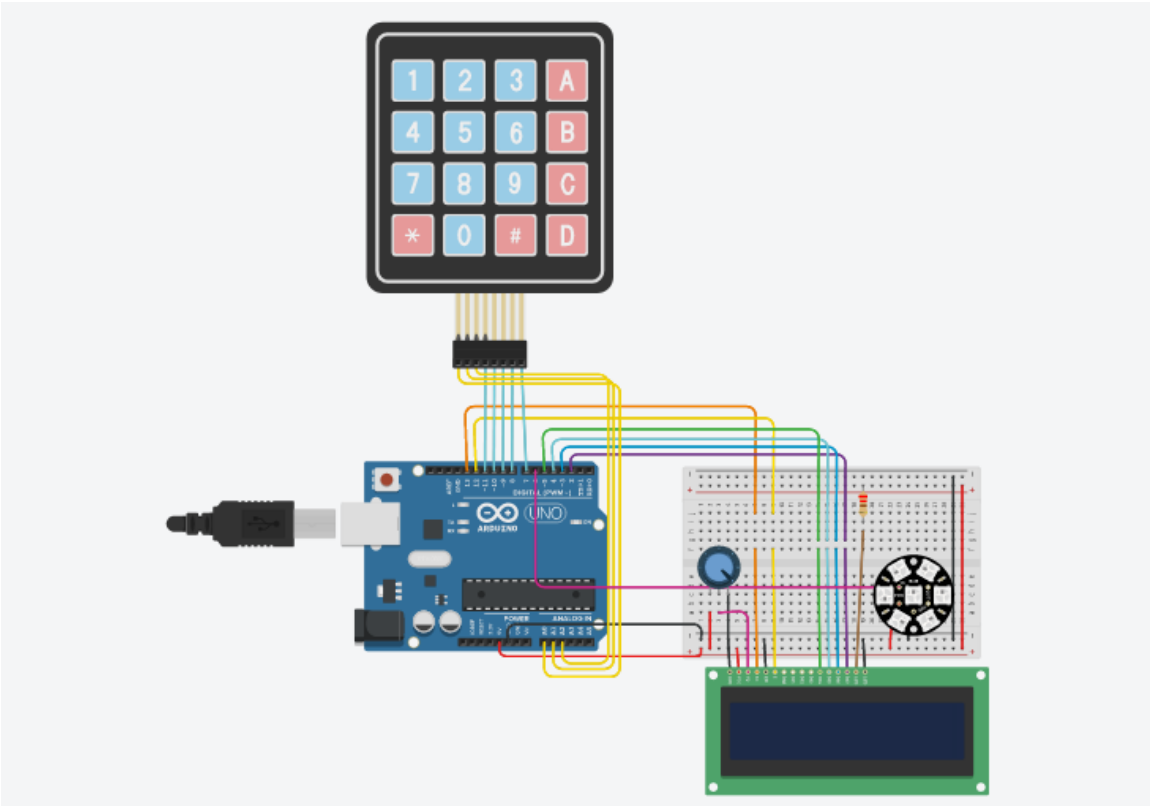

Fig.3.1 Schematic diagram (Tinkercad)

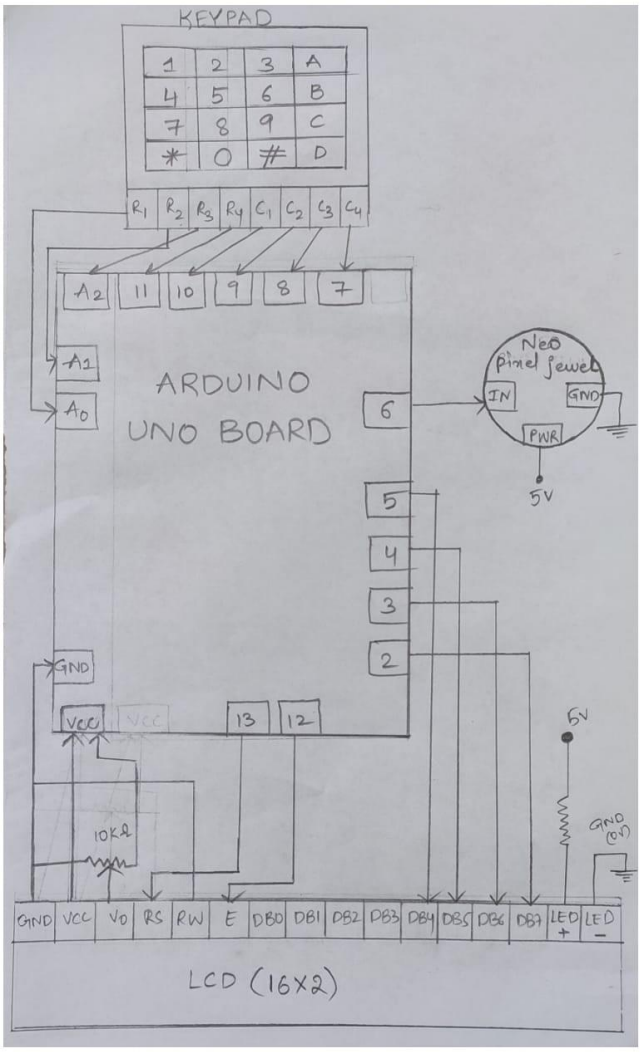

Fig.3.2 Schematic diagram (Drawn)

3.2 RESULTS

The experimental results are as follows.

- When the program is executed, we get a message on LCD as shown in figure 3.3.

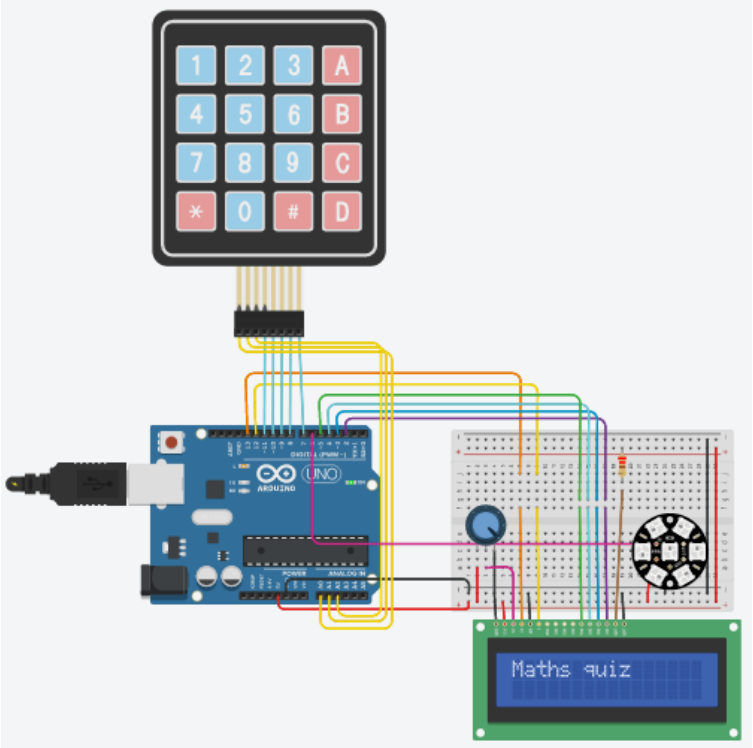

Fig.3.3 Result_1

- After 1 second, as shown in figure 3.4, we display a short menu with different operations.

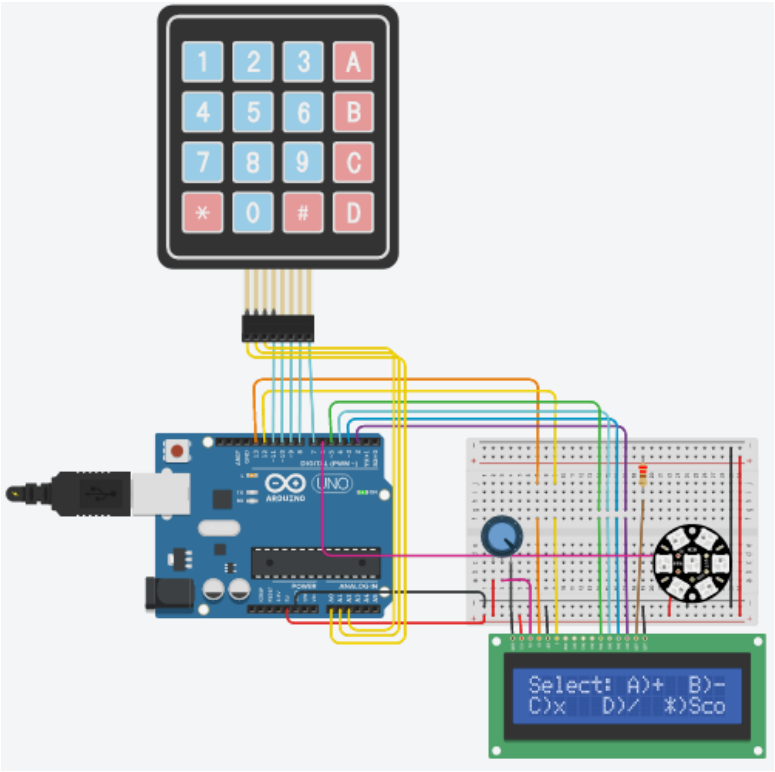

Fig.3.4 Result_2

- The user can choose the desired operation through the keypad. Then their selected operation is displayed on the LCD as shown in the figure below.

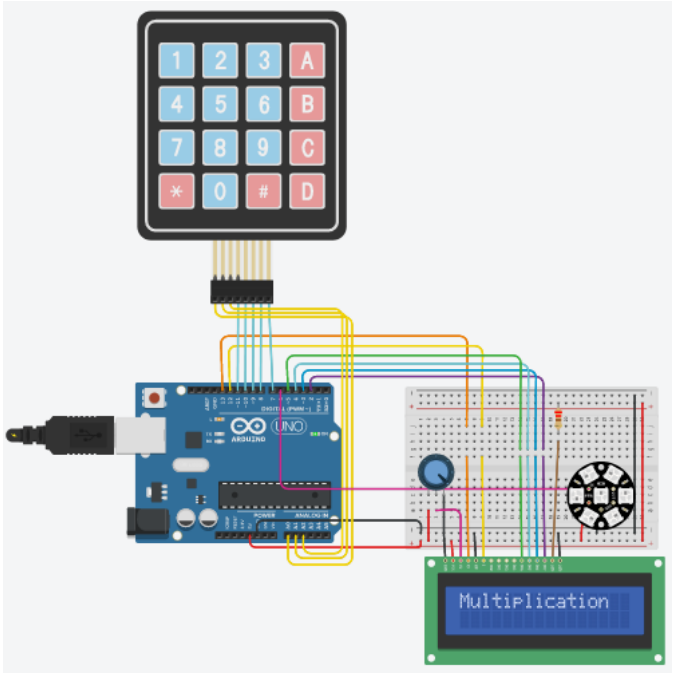

Fig.3.5 Result_3

- Based on the user's choice of operation, the question is displayed on the LCD which can be seen in the below figure 3.6

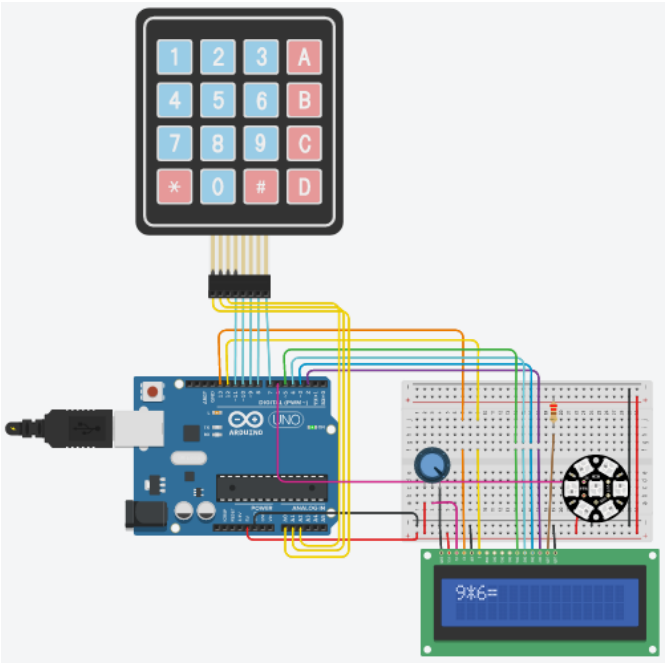

Fig.3.6 Result_4

- The user attempts to give answer to the math question displayed on the LCD. The input can be given through the keypad. As we can see in the below figure 3.7, the answer is printed on the LCD.

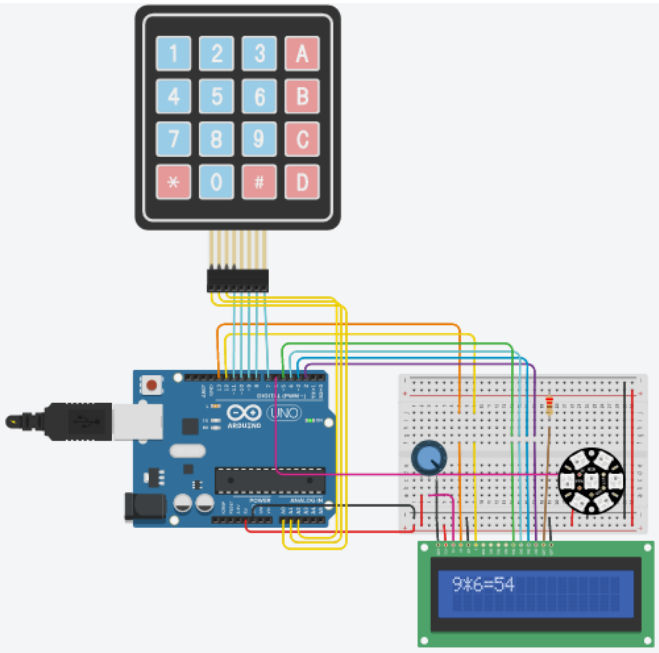

Fig.3.7 Result_5

- When the user submits the answer, if their answer is correct, a message is displayed as shown in the figure 3.8 and the neo pixel gives green light.

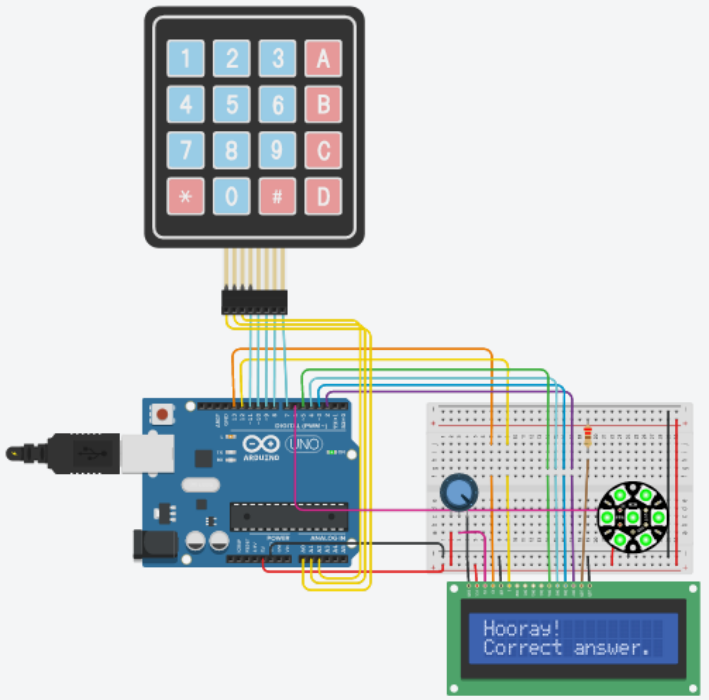

Fig.3.8 Result_6

- After a few seconds, the menu is displayed again, from which the user can choose any operation or to view their score. Here, we are selecting to solve another question on addition.

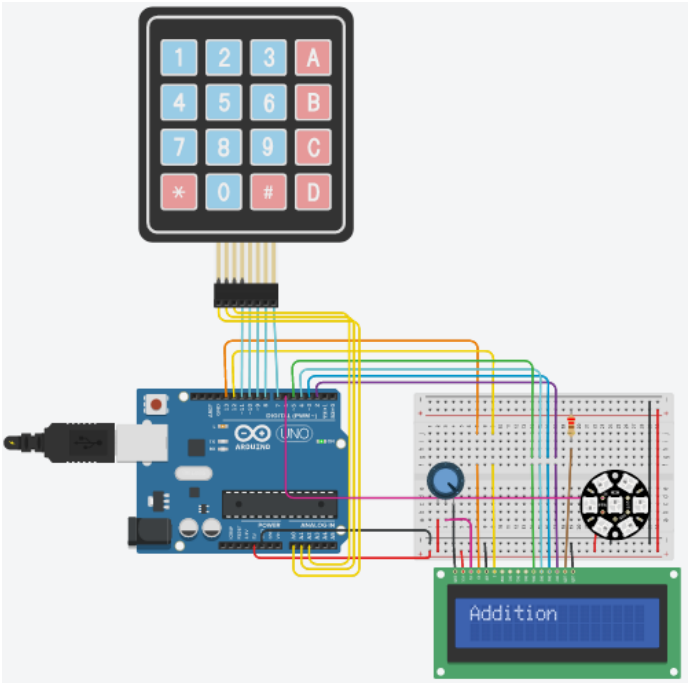

Fig.3.9 Result_7

- The question is displayed according to the selected operation, for which the user can give answer through the keypad.

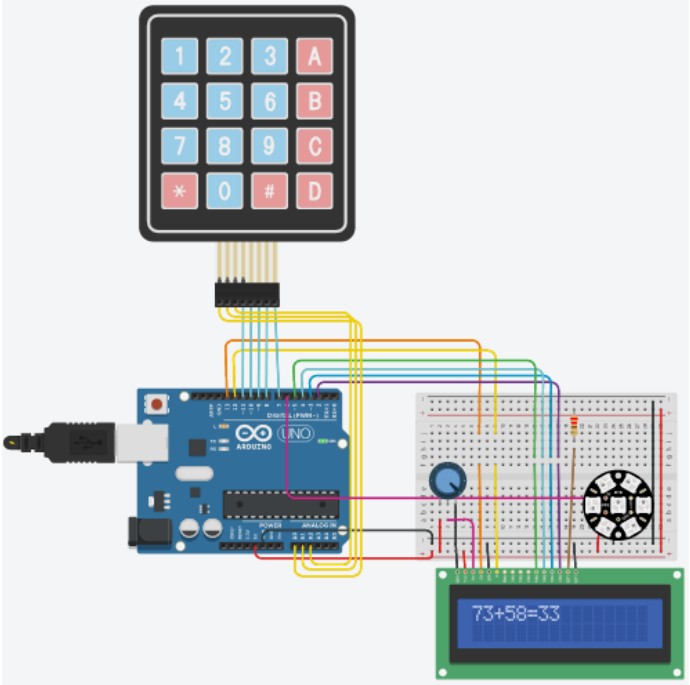

Fig.3.10 Result_8

- When the user submits the answer, if their answer is wrong, a message is displayed as shown in the figure 3.11 and the correct answer is also displayed. Meanwhile, the neo pixel gives red light.

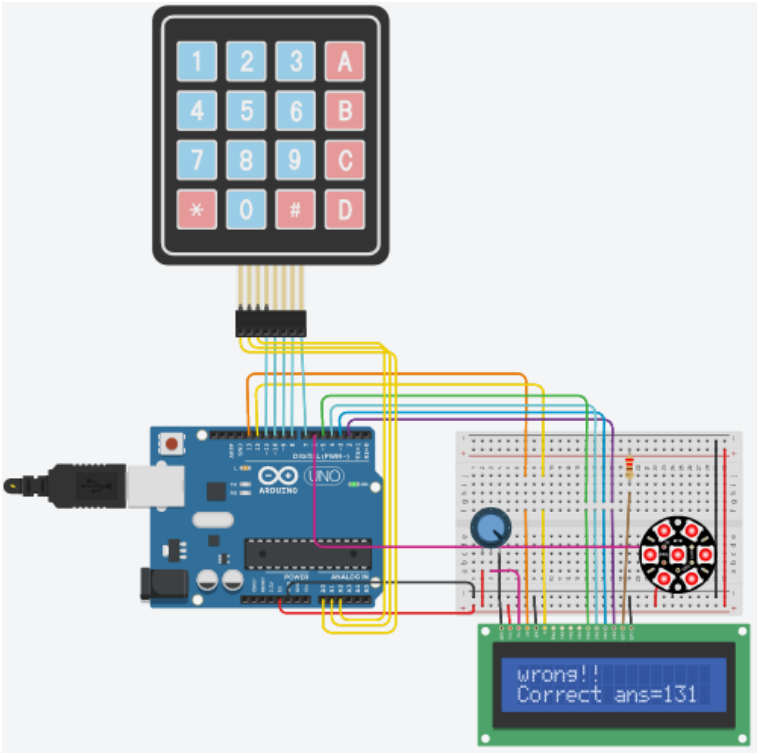

Fig.3.11 Result_9

- After a few seconds the menu is displayed again. Here we are choosing to view the score. As shown in the below figure 3.12, the score is displayed. Here, we got the score as ½. This means that out of two questions one answer is correct. The neo pixel jewel shows yellow light while displaying the score.

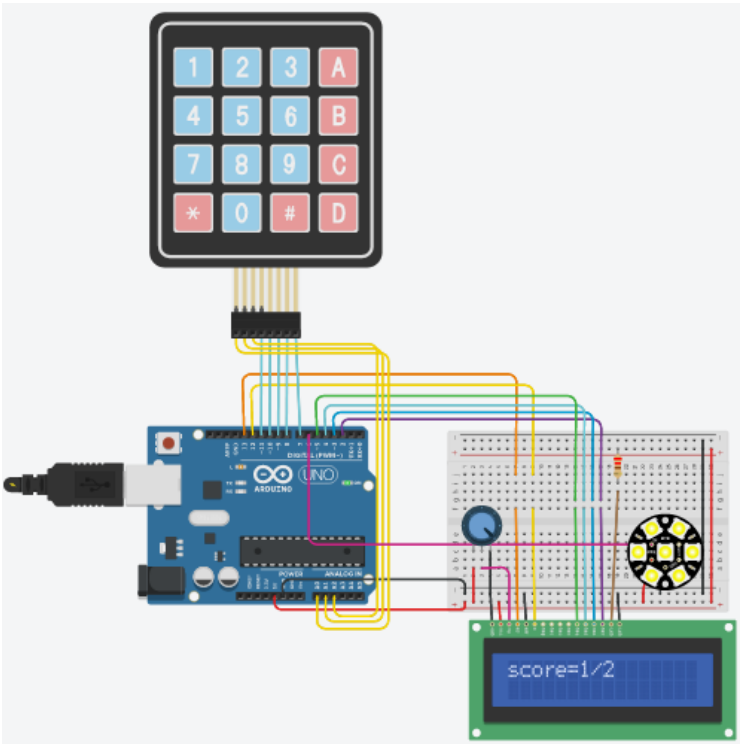

Fig.3.12 Result_10

- Then again, after some time, the score is reset to zero and quiz starts again from the beginning.

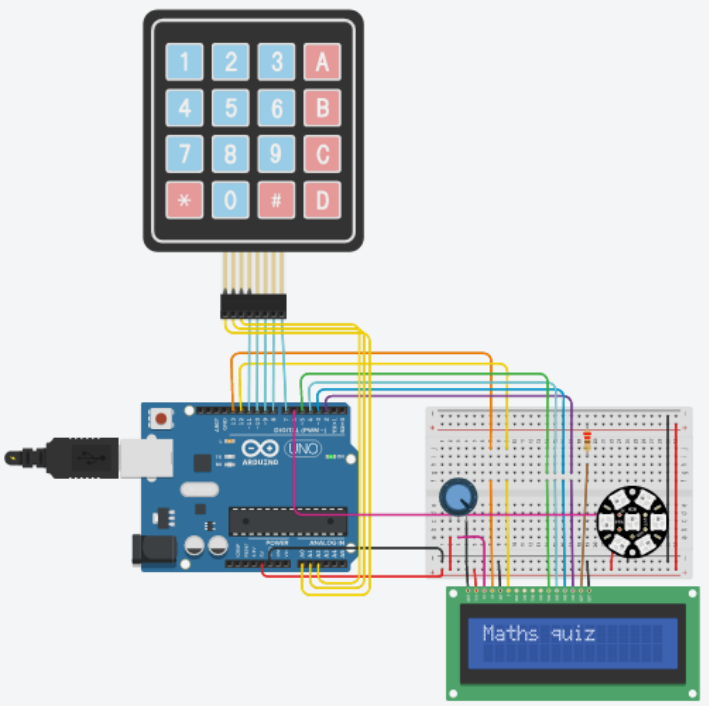

Fig.3.13 Result_11

3.3 ADVANTAGES

The advantages of mathematics smart quiz are:

- The quiz is interactive so, it can keep the interest of the user and helps them to indulge in the experience of learning. It is a play way method of learning.
- Questions will be asked based on the mathematical operation the student chooses.
- If the user gives wrong answer, correct answer is displayed for their reference. So that they can learn and not repeat that mistake again.
- As there is no time limit, the student can feel less stressful to answer the question and they can think thoroughly before giving their answer.
- Score can be viewed whenever the user wishes to, and they can start the quiz from the beginning, again.
- The questions are generated automatically, so we cannot predict which question comes next.
- The quiz is color coded, that is, the neo pixel jewel shows green and red colors based on the response of the student.

3.4 DISADVANTAGES

- The quiz covers questions only from 4 basic mathematical operations.
- Limited number of keys on keypad.

CHAPTER 4

CONCLUSION

4.1 CONCLUSION

To sum it up, our project, mathematics smart quiz makes ‘practice feel like play’. In our smart quiz, the Arduino poses math questions to the user based on the operation they select and checks their response. Through our project we test and challenge the problem solving abilities of the students.

4.2 FUTURE SCOPE

Our product can be further improved by adding some additional features, a few of them are mentioned below.

- More mathematical operations can be included
- We can create a reward system when a person gives correct answer.
- We can include a better keypad with more keys.
- We can include different modes of quiz like relaxed mode and timed mode
- We can store scores of different players and create a high scores leaderboard.

BIBLIOGRAPHY

[1]. Learn electronics with Arduino(2017) by Jody culkin and Eric hagan

[2].Getting started with Arduino, 3rd edition(2009) by Massimo Banzi

[3]. https://www.electroschematics.com/arduino-with-keypad/

[4]. https://components101.com/displays/neopixel-led-strip

[5].C programming for Arduino(2013) by Julien Bayle

APPENDIX

```
#include <Adafruit_NeoPixel.h>
#define PIN 6   // input pin Neopixel is attached to
#define NUMPIXELS 7 // number of neopixels in strip
#include <Keypad.h>
#include<LiquidCrystal.h>
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
LiquidCrystal lcd (13,12,5,4,3,2);
const byte numRows=4;
const byte numCols=4;
char keymap[numRows][numCols]=
{
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}};
byte rowPins[numRows]={A0, A1, A2, 11};
byte colPins[numCols]={10,9,8,7};
Keypad myKeypad=Keypad(makeKeymap(keymap),rowPins, colPins, numRows,numCols);
int a,b,c,count=0,total=0,n=0;
void setup()
{
  pixels.begin();
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("Maths quiz");
  delay(700);
}
void loop()
{
  pixels.setBrightness(255);
  char key=myKeypad.getKey();
  int num=0;
```

```
lcd.setCursor(0,1);
  lcd.clear();
  lcd.print("Select: A)+  B)-");
  lcd.setCursor(0,1);
  lcd.print("C)x   D)/  *)Score");
  delay(6000);
  lcd.clear();
while(key != '#')
 {
    switch (key)
    {
      case NO_KEY:
                          break;
      case '0':
      case '1':
      case '2':
      case '3':
      case '4':
      case '5':
      case '6':
      case '7':
      case '8':
      case '9':
                          lcd.print(key);
                          num = num * 10 + (key - '0');
                          break;

      case '*':  lcd.clear();
                                lcd.print("score=");
                                lcd.print(count);
                          lcd.print("/");
                                lcd.print(total);
                          for(int i=0;i<7;i++)
               pixels.setPixelColor(i, pixels.Color(255, 255, 0));
```

```
        pixels.show();
        delay(4000);
                for(int i=0;i<7;i++)
                        pixels.setPixelColor(i, pixels.Color(0, 0, 0));
                        pixels.show();
                        lcd.clear();
                count=0;
                total=0;
                        lcd.print("Thank you!");
                        delay(3000);
                lcd.clear();
                lcd.print("Maths quiz");
                delay(2000);
                lcd.clear();
                lcd.print("Select: A)+  B)-");
                        lcd.setCursor(0,1);
                                lcd.print("C)x   D)/  *)Score");
                        delay(3000);
            break;
case 'A':  lcd.print("Addition");
                delay(700);
                a=random(100);
                b=random(100);
                c=a+b;
                                lcd.clear();
                                lcd.setCursor(0,0);
                                lcd.print(a);
                                lcd.print("+");
                                lcd.print(b);
                                lcd.print("=");
   break;
case 'B':  lcd.print("Subtraction");
                delay(700);
                a=random(60,100);
```

```
                    b=random(11,50);
                    c=a-b;
                                    lcd.clear();
                                    lcd.setCursor(0,0);
                                    lcd.print(a);
                                    lcd.print("-");
                                    lcd.print(b);
                                    lcd.print("=");
    break;
 case 'C':  lcd.print("Multiplication");
                    delay(700);
                    a=random(2,12);
                    b=random(5,11);
                    c=a*b;
                                    lcd.clear();
                                    lcd.setCursor(0,0);
                                    lcd.print(a);
                                    lcd.print("*");
                                    lcd.print(b);
                                    lcd.print("=");
    break;
 case 'D':  lcd.print("Division");
                    delay(700);
                    a=random(9,20);
                    b=random(9);
                    c=a/b;
                                    lcd.clear();
                                    lcd.setCursor(0,0);
                                    lcd.print(a);
                                    lcd.print("/");
                                    lcd.print(b);
                                    lcd.print("=");
    break;
}
```

```
 key=myKeypad.getKey();
}
 Serial.print(num);
 if(c==num)
 {
  lcd.clear();
  lcd.print("Hooray!");
  lcd.setCursor(0,1);
  lcd.print("Correct answer.");
  for(int i=0;i<7;i++)
  pixels.setPixelColor(i, pixels.Color(0, 255, 0));
  pixels.show();
  delay(2500);
  for(int i=0;i<7;i++)
  pixels.setPixelColor(i, pixels.Color(0, 0, 0));
  pixels.show();
  count++;
  total++;
 }
       else
 {
  lcd.clear();
  lcd.print("wrong!!");
  lcd.setCursor(0,1);
  lcd.print("Correct ans=");
  lcd.print(c);
  for(int i=0;i<7;i++)
  pixels.setPixelColor(i, pixels.Color(255, 0, 0));
  pixels.show();
  delay(2500);
  for(int i=0;i<7;i++)
  pixels.setPixelColor(i, pixels.Color(0, 0, 0));
  pixels.show();
  total++;
```

```
    }
}
```