

```

#include <Keypad.h>

#define RELAY_PIN  A0 // the Arduino pin that controls solenoid lock
via relay
#define ROW_NUM    4  // four rows
#define COLUMN_NUM 4  // four columns

char keys[ROW_NUM][COLUMN_NUM] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte pin_rows[ROW_NUM] = {9, 8, 7, 6}; // connect to the row pinouts of
the keypad
byte pin_column[COLUMN_NUM] = {5, 4, 3, 2}; // connect to the column
pinouts of the keypad

Keypad keypad = Keypad( makeKeymap(keys), pin_rows, pin_column,
ROW_NUM, COLUMN_NUM );

const String password_1 = "1234"; // change your password here
const String password_2 = "56789"; // change your password here
const String password_3 = "901234"; // change your password here
String input_password;

unsigned long unlockStartTime = 0;
const unsigned long unlockDuration = 5000; // 3 seconds

void setup() {
    Serial.begin(9600);
    input_password.reserve(32); // maximum password size is 32, change if
needed
    pinMode(RELAY_PIN, OUTPUT); // initialize pin as an output.
    digitalWrite(RELAY_PIN, HIGH); // lock the solenoid lock
}

void loop() {
    char key = keypad.getKey();

    if (key) {
        Serial.println(key);
    }
}

```

```

    if (key == '*') {
        input_password = ""; // reset the input password
    } else if (key == '#') {
        if (input_password == password_1 || input_password == password_2
|| input_password == password_3) {
            Serial.println("The password is correct => unlock");
            digitalWrite(RELAY_PIN, LOW);
            unlockStartTime = millis(); // record the start time of the
unlock process
        } else {
            Serial.println("The password is incorrect, try again");
        }

        input_password = ""; // reset the input password
    } else {
        input_password += key; // append new character to input password
string
    }
}

// Check if it's time to lock again
if (millis() - unlockStartTime >= unlockDuration) {
    digitalWrite(RELAY_PIN, HIGH); // lock the solenoid lock
}
}

```