1). WRITE AA PROGRAM TO BLINK AN LED WHEN A BUTTON 1 IS PRESSED % TURN OFF AN LED WHEN BUTTON 2 IS PRESSED? DEVELOP CODE USING ARM MICRO CONTROLLER?

```
#include "LPC17xx.h"
#define LED_PIN    (1 << 0)  // P1.0 for LED
#define BUTTON1    (1 << 0)  // P2.0 for Button 1
#define BUTTON2    (1 << 1)  // P2.1 for Button 2

void delay_ms(uint32_t ms) {
    uint32_t i, j;
    for (i = 0; i < ms; i++) {
        for (j = 0; j < 5000; j++); // Approximate delay
    }
}

int main(void) {
    // Configure LED pin as output
    LPC_GPIO1->FIODIR |= LED_PIN;

    // Configure Button 1 and Button 2 as input
    LPC_GPIO2->FIODIR &= ~(BUTTON1 | BUTTON2);

    while (1) {
        if (!(LPC_GPIO2->FIOPIN & BUTTON1)) { // If Button 1 is pressed
            LPC_GPIO1->FIOSET = LED_PIN;  // Turn ON LED
            delay_ms(500); // Blink effect
            LPC_GPIO1->FIOCLR = LED_PIN;  // Turn OFF LED
            delay_ms(500);
        }
        if (!(LPC_GPIO2->FIOPIN & BUTTON2)) { // If Button 2 is pressed
            LPC_GPIO1->FIOCLR = LED_PIN;  // Turn OFF LED
        }
    }
}
```

2) . WRITE A PROGRAM TO ROTATE DC MOTOR CLOCK WISE WHEN BUTTON 1 IS PRESSED & ANTI CLOCK WISE WHEN BUTTON 2 IS PRESSED ? DEVELOP A CODE USING ARM MICRO CONTROLLER ?

```
#include "LPC17xx.h"
#define MOTOR_IN1 (1 << 0)  // Motor IN1 -> P1.0
#define MOTOR_IN2 (1 << 1)  // Motor IN2 -> P1.1
#define BUTTON1   (1 << 0)  // Button 1 -> P2.0
#define BUTTON2   (1 << 1)  // Button 2 -> P2.1

void delay_ms(uint32_t ms) {
    uint32_t i, j;
    for (i = 0; i < ms; i++) {
        for (j = 0; j < 5000; j++); // Approximate delay
    }
}

int main(void) {
    // Configure Motor Control Pins as Output
    LPC_GPIO1->FIODIR |= (MOTOR_IN1 | MOTOR_IN2);

    // Configure Buttons as Input
    LPC_GPIO2->FIODIR &= ~(BUTTON1 | BUTTON2);

    while (1) {
        if (!(LPC_GPIO2->FIOPIN & BUTTON1)) {  // If Button 1 is pressed
            LPC_GPIO1->FIOSET = MOTOR_IN1;  // Turn ON IN1 (Clockwise)
            LPC_GPIO1->FIOCLR = MOTOR_IN2;  // Turn OFF IN2
        }
        else if (!(LPC_GPIO2->FIOPIN & BUTTON2)) {  // If Button 2 is pressed
```

```
            LPC_GPIO1->FIOSET = MOTOR_IN2;  // Turn ON IN2 (Anti-clockwise)
            LPC_GPIO1->FIOCLR = MOTOR_IN1;  // Turn OFF IN1
        }
        else {  // No button pressed, stop motor
            LPC_GPIO1->FIOCLR = MOTOR_IN1 | MOTOR_IN2;
        }

        delay_ms(10); // Small delay to debounce buttons
    }
}
```

3). CONTROL A DC MOTOR USING 2 BUTTONS & DISPLAY THE MOTOR DIRECTIONS ON LCD SCREEN . DEVELOP USING ARDUINO UNO MICRO CONTROLLER?

```
#include <LiquidCrystal_I2C.h>
#define MOTOR_IN1 6  // Motor Driver Input 1
#define MOTOR_IN2 7  // Motor Driver Input 2
#define BUTTON1 2    // Button 1 (Clockwise)
#define BUTTON2 3    // Button 2 (Anti-clockwise)

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
    pinMode(MOTOR_IN1, OUTPUT);
    pinMode(MOTOR_IN2, OUTPUT);
    pinMode(BUTTON1, INPUT_PULLUP);
    pinMode(BUTTON2, INPUT_PULLUP);

    lcd.begin();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("DC Motor Control");
    delay(1000);
    lcd.clear();
}

void loop() {
    if (digitalRead(BUTTON1) == LOW) {  // Button 1 pressed
        digitalWrite(MOTOR_IN1, HIGH);
        digitalWrite(MOTOR_IN2, LOW);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Motor: Clockwise");
    }
    else if (digitalRead(BUTTON2) == LOW) {  // Button 2 pressed
        digitalWrite(MOTOR_IN1, LOW);
        digitalWrite(MOTOR_IN2, HIGH);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Motor: Anti-Clock");
    }
    else {  // No button pressed -> Stop Motor
        digitalWrite(MOTOR_IN1, LOW);
        digitalWrite(MOTOR_IN2, LOW);
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Motor: Stopped");
    }
    delay(200);  // Small delay for debounce
}
```

4). BLINK 5 LED IN SERIES USING RASPBERRY PI CONTROLLER ?

```
import RPi.GPIO as GPIO
import time
# Define LED GPIO pins
```

```
LED_PINS = [17, 18, 27, 22, 23]


# Setup
GPIO.setmode(GPIO.BCM)  # Use Broadcom GPIO numbering
GPIO.setup(LED_PINS, GPIO.OUT)


try:
   while True:
      # Turn LEDs ON one by one
      for led in LED_PINS:
         GPIO.output(led, GPIO.HIGH)
         time.sleep(0.5)  # 500ms delay


      # Turn LEDs OFF one by one
      for led in LED_PINS:
         GPIO.output(led, GPIO.LOW)
         time.sleep(0.5)


except KeyboardInterrupt:
   print("Exiting program...")
   GPIO.cleanup()  # Reset GPIO settings
B
```

5). CHECK IF A GIVEN MOBILE NUMBER IS CORRECT USING MAT LAB ?

```
a=input(“enter valid mobile number”)
if(len(a)==10):
    If((a[0]==9) |( a[0]==8)|(a[0]==7)|(a[0]==6)):
               display (“entered number is valid”)
else:
```

```
display(“invalid”)
```