HOMEWORK MICROPROCESSOR 1:

Task1:

void setup() {

// Set PB2 to PB5 (digital pins 10 to 13) as outputs

DDRB |= (1 << DDB2) | (1 << DDB3) | (1 << DDB4) | (1 << DDB5);

}

void loop() {

// Binary sequence to display: 1, 3, 7, 15

uint8\_t sequence[] = {0b0001, 0b0011, 0b0111, 0b1111};

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

// Clear PB2 to PB5

PORTB &= ~((1 << PORTB2) | (1 << PORTB3) | (1 << PORTB4) | (1 << PORTB5));

// Set LEDs based on the sequence (shifted to align with PB2–PB5)

PORTB |= (sequence[i] << 2);

// Delay between numbers (approx. 500ms)

delay\_ms(1000);

}

}

// Simple delay function (busy wait loop)

void delay\_ms(unsigned int ms) {

for (unsigned int i = 0; i < ms; i++) {

for (volatile unsigned int j = 0; j < 1000; j++);

}

}

TASK2:

void setup() {

// Set PB1 (pin 9) as output for red LED

DDRB |= (1 << DDB1);

// Set PB0 (pin 8) as input for button

DDRB &= ~(1 << DDB0);

// Enable pull-up resistor on PB0 (pin 8)

PORTB |= (1 << PORTB0);

}

void loop() {

// Check if button is pressed (reads LOW)

if (!(PINB & (1 << PINB0))) {

// Button pressed → turn OFF red LED

PORTB &= ~(1 << PORTB1);

} else {

// Button not pressed → red LED ON

PORTB |= (1 << PORTB1);

}

}

TASK3:

// Segment-to-pin definitions

#define SEG\_A 7

#define SEG\_B 6

#define SEG\_C 5

#define SEG\_D 4

#define SEG\_E 3

#define SEG\_F 2

#define SEG\_G 1

#define SEG\_DP 0 // Decimal point

const uint8\_t segmentPins[8] = {

SEG\_A, SEG\_B, SEG\_C, SEG\_D, SEG\_E, SEG\_F, SEG\_G, SEG\_DP

};

// Segment patterns for 0–9 (A B C D E F G DP)

const uint8\_t digitSegments[10][8] = {

{1,1,1,1,1,1,0,0}, // 0

{0,1,1,0,0,0,0,0}, // 1

{1,1,0,1,1,0,1,0}, // 2

{1,1,1,1,0,0,1,0}, // 3

{0,1,1,0,0,1,1,0}, // 4

{1,0,1,1,0,1,1,0}, // 5

{1,0,1,1,1,1,1,0}, // 6

{1,1,1,0,0,0,0,0}, // 7

{1,1,1,1,1,1,1,0}, // 8

{1,1,1,1,0,1,1,0} // 9

};

void setup() {

for (int i = 0; i < 8; i++) {

setPinMode(segmentPins[i], 1); // 1 = OUTPUT

}

}

void loop() {

for (int digit = 0; digit <= 9; digit++) {

displayDigit(digit);

delay\_ms(500);

}

}

// Display digit by setting each segment

void displayDigit(uint8\_t digit) {

for (int i = 0; i < 8; i++) {

digitalWrite(segmentPins[i], digitSegments[digit][i]);

}

}

// Set pin as OUTPUT (only PORTD pins 0–7 used)

void setPinMode(uint8\_t pin, uint8\_t mode) {

if (mode)

DDRD |= (1 << pin);

else

DDRD &= ~(1 << pin);

}

// Set pin HIGH/LOW

void digitalWrite(uint8\_t pin, uint8\_t value) {

if (value)

PORTD |= (1 << pin);

else

PORTD &= ~(1 << pin);

}

// Rough millisecond delay

void delay\_ms(unsigned int ms) {

for (unsigned int i = 0; i < ms; i++) {

for (volatile unsigned int j = 0; j < 1000; j++);

}

}