**2nd Lab - General Purpose Input/Output Based Input and Output Operations Using 8051 Microcontroller**

**Implementation:**

**Keil:**

**1st Project:**

Open Keil µVision → Go to Project → New µVision Project → Set a project name and save it → In the Select Device for Target window, search and select AT89C51 → Click OK → When prompted, click No → In Project window, right-click Source Group 1 under Target 1 → Select Add New Item to Group 'Source Group 1' → Create and save a new C file → Write the code below and save:

**Source Code:**

#include<reg51.h>

void Delay (unsigned int time);

void main()

{

P2 = 0xAA;

Delay(1000);

P2=0x55;

Delay(1000);

}

void Delay (unsigned int time)

{

unsigned int i,j;

for (i=0; i<time; i++)

{

for (j=0; j<23; j++) {}

}

}

→ Right-click Source Group 1 → Add Existing Files to Group → Select the saved .c file → Click Add and Close → Right-click Target 1 → Options for Target → Go to Output tab → Check "Create Hex File" → Go to Target tab → Set Xtal (MHz) to 11.0592 → Click OK → Press F7 to build the project → If there are no errors or warnings, your code is okay.

**2nd Project:**

Open Keil µVision → Go to Project → New µVision Project → Set a project name and save it → In the Select Device for Target window, search and select AT89C51 → Click OK → When prompted, click No → In Project window, right-click Source Group 1 under Target 1 → Select Add New Item to Group 'Source Group 1' → Create and save a new C file → Write the code below and save:

**Source Code:**

#include<reg51.h>

void Delay (unsigned int time);

int main()

{

while (1)

{

P2=P1;

Delay(1000);

}

}

void Delay (unsigned int time)

{

unsigned int i,j;

for(i=0; i<time; i++)

{

for(j=0; j<23; j++) {}

}

}

→ Right-click Source Group 1 → Add Existing Files to Group → Select the saved .c file → Click Add and Close → Right-click Target 1 → Options for Target → Go to Output tab → Check "Create Hex File" → Go to Target tab → Set Xtal (MHz) to 11.0592 → Click OK → Press F7 to build the project → If there are no errors or warnings, your code is okay.

**Proteus:**

Open Proteus → Click New Project → Set project name → Keep clicking Next until Finish → Click Finish → From the left sidebar, select Component Mode → Click P → Search and add AT89C51 ×2 → Add LED ×8 → From the sidebar, select Terminals Mode → Add Ground

**Component Placement:**

Place U1 (first AT89C51) and U2 (second AT89C51) on the workspace → Place 8 LEDs (D1–D8) horizontally at the bottom → Place Ground symbol near LED cathode side

**Wiring:**

Connect U1 P2.0–P2.7 (pins 21–28) to U2 P1.0–P1.7 (pins 1–8) → Connect U2 P2.0–P2.7 (pins 21–28) to anode side of 8 LEDs → Connect cathode side of all LEDs to Ground

**Programming:**

Double-click U1 → Set clock frequency = 11.0592 MHz → Browse and select U1's HEX file → Click OK → Double-click U2 → Set clock frequency = 11.0592 MHz → Browse and select U2's HEX file → Click OK

**Simulation:**

Click the Play button → If LEDs blink correctly, your setup is working → Save the project

**Diagram:**