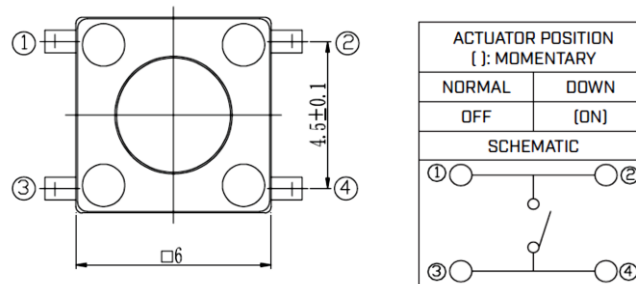**EE 2305 – Introduction to C Programming**
**Hardware Project 02**
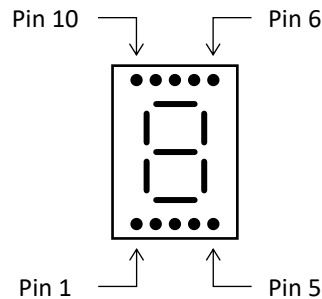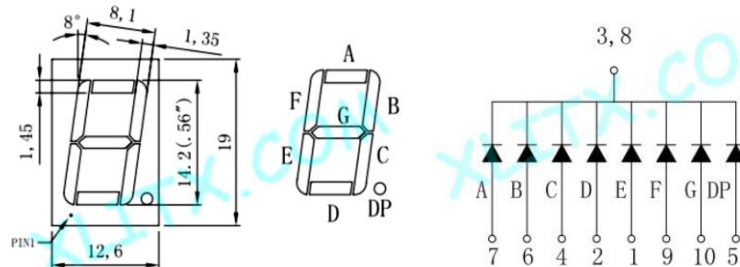
**7-Segment Display**

Project Features:    Digital Input and Digital Output.

Program an Arduino board to accept a 4-bit binary input and display the numerical value (0 to 9) on a 7-Segment LED display.
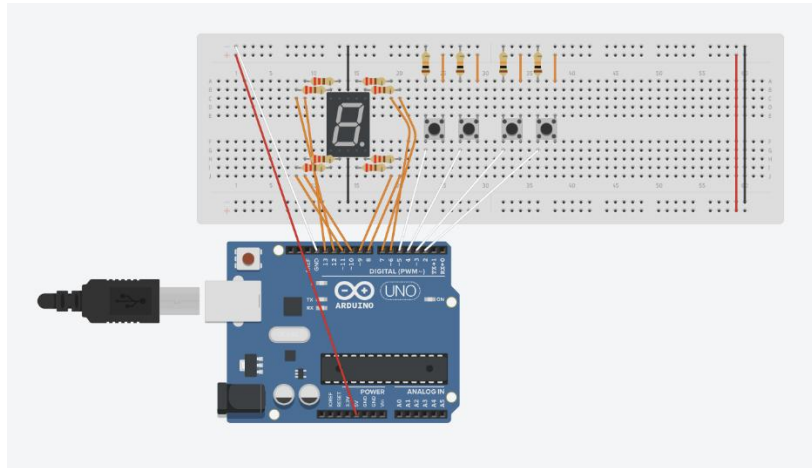
Use the pushbutton for the digital input:



Use the 7-Segment LED Display:



To document your program, create a document and include the following sections in the document. Provide a brief description of the system and how you are designing it to operate,
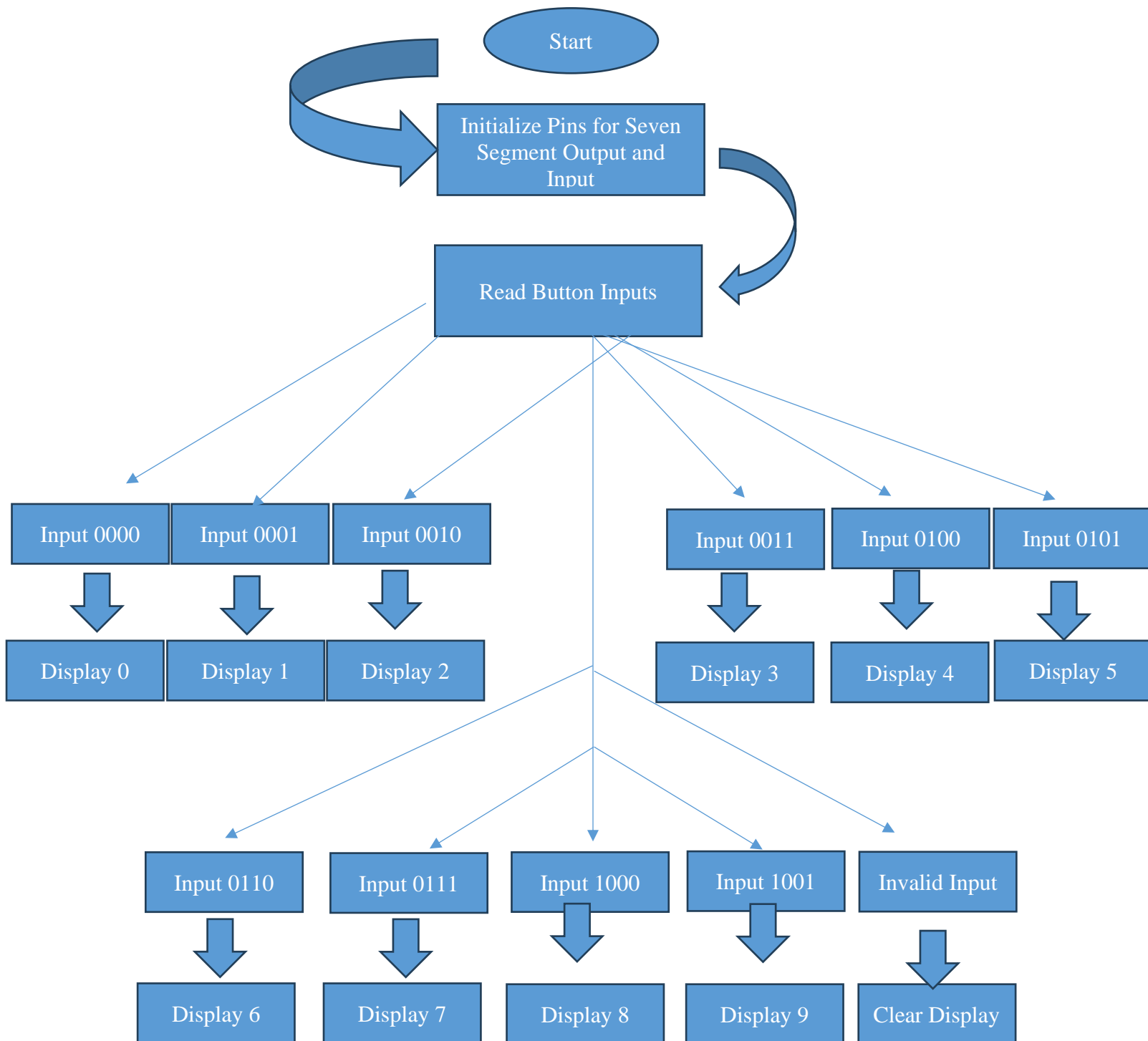
## A. Hardware Diagram:

Provide a hardware diagram of the components.

**B. Program Flowchart:**

Draw a flowchart of the program using the Word graphics shapes.

```
                          ┌─────────────┐
                          │    Start    │
                          └─────────────┘

              ┌──────────────────────────┐
              │ Initialize Pins for Seven │
              │   Segment Output and      │
              │         Input             │
              └──────────────────────────┘

              ┌──────────────────────────┐
              │    Read Button Inputs     │
              └──────────────────────────┘
```

| Input 0000 | Input 0001 | Input 0010 |   | Input 0011 | Input 0100 | Input 0101 |
|------------|------------|------------|---|------------|------------|------------|
| Display 0  | Display 1  | Display 2  |   | Display 3  | Display 4  | Display 5  |

| Input 0110 | Input 0111 | Input 1000 | Input 1001 | Invalid Input |
|------------|------------|------------|------------|---------------|
| Display 6  | Display 7  | Display 8  | Display 9  | Clear Display |

## C. Arduino Source Code

```cpp
void setup()
{
  // Setup output pins for the 7-segment display
  pinMode(12, OUTPUT); // segment g
  pinMode(13, OUTPUT); // segment f
  pinMode(11, OUTPUT); // segment e
  pinMode(10, OUTPUT); // segment d
  pinMode(9, OUTPUT); // segment c
  pinMode(7, OUTPUT); // segment b
  pinMode(6, OUTPUT); // segment a
  // Setup input pins with internal pull-up resistors for the buttons (bit 3 to
bit 0)
  pinMode(5, INPUT_PULLUP); // bit 3
  pinMode(4, INPUT_PULLUP); // bit 2
  pinMode(3, INPUT_PULLUP); // bit 1
  pinMode(2, INPUT_PULLUP); // bit 0
}
  void loop()
  {
    // Read binary input from buttons (inverse logic due to INPUT_PULLUP)
    if (!digitalRead(5) && !digitalRead(4) && !digitalRead(3) && !digitalRead(2))
    {
      // Display 0
      digitalWrite(13, HIGH);
      digitalWrite(12, LOW);
      digitalWrite(11, HIGH);
      digitalWrite(10, HIGH);
      digitalWrite(9, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(6, HIGH);
    }
    else if (!digitalRead(5) && !digitalRead(4) && !digitalRead(3) &&
digitalRead(2))
    {
      // Display 1
      digitalWrite(13, LOW);
      digitalWrite(12, LOW);
      digitalWrite(11, LOW);
      digitalWrite(10, LOW);
      digitalWrite(9, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(6, LOW);
    }
```

```
    else if (!digitalRead(5) && !digitalRead(4) && digitalRead(3) &&
!digitalRead(2))
    {
      // Display 2
      digitalWrite(13, LOW);
      digitalWrite(12, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(10, HIGH);
      digitalWrite(9, LOW);
      digitalWrite(7, HIGH);
      digitalWrite(6, HIGH);
    }
    else if (!digitalRead(5) && !digitalRead(4) && digitalRead(3) &&
digitalRead(2))
    {
      // Display 3
      digitalWrite(13, LOW);
      digitalWrite(12, HIGH);
      digitalWrite(11, LOW);
      digitalWrite(10, HIGH);
      digitalWrite(9, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(6, HIGH);
    }
    else if (!digitalRead(5) && digitalRead(4) && !digitalRead(3) &&
!digitalRead(2))
    {
      // Display 4
      digitalWrite(13, HIGH);
      digitalWrite(12, HIGH);
      digitalWrite(11, LOW);
      digitalWrite(10, LOW);
      digitalWrite(9, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(6, LOW);
    }
    else if (!digitalRead(5) && digitalRead(4) && !digitalRead(3) &&
digitalRead(2))
    {
      // Display 5
      digitalWrite(13, HIGH);
      digitalWrite(12, HIGH);
      digitalWrite(11, LOW);
      digitalWrite(10, HIGH);
      digitalWrite(9, HIGH);
```

```
      digitalWrite(7, LOW);
      digitalWrite(6, HIGH);
    }
    else if (!digitalRead(5) && digitalRead(4) && digitalRead(3) &&
!digitalRead(2))
    {
      // Display 6
      digitalWrite(13, HIGH);
      digitalWrite(12, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(10, HIGH);
      digitalWrite(9, HIGH);
      digitalWrite(7, LOW);
      digitalWrite(6, HIGH);
    }
    else if (!digitalRead(5) && digitalRead(4) && digitalRead(3) &&
digitalRead(2))
    {
      // Display 7
      digitalWrite(13, LOW);
      digitalWrite(12, LOW);
      digitalWrite(11, LOW);
      digitalWrite(10, LOW);
      digitalWrite(9, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(6, HIGH);
    }
    else if (digitalRead(5) && !digitalRead(4) && !digitalRead(3) &&
!digitalRead(2))
    {
      // Display 8
      digitalWrite(13, HIGH);
      digitalWrite(12, HIGH);
      digitalWrite(11, HIGH);
      digitalWrite(10, HIGH);
      digitalWrite(9, HIGH);
      digitalWrite(7, HIGH);
      digitalWrite(6, HIGH);
    }
    else if (digitalRead(5) && !digitalRead(4) && !digitalRead(3) &&
digitalRead(2))
    {
      // Display 9
      digitalWrite(13, HIGH);
      digitalWrite(12, HIGH);
```

```
    digitalWrite(11, LOW);
    digitalWrite(10, HIGH);
    digitalWrite(9, HIGH);
    digitalWrite(7, HIGH);
    digitalWrite(6, HIGH);
  }
  else
  {
    // Clear display (no valid input)
    digitalWrite(13, LOW);
    digitalWrite(12, LOW);
    digitalWrite(11, LOW);
    digitalWrite(10, LOW);
    digitalWrite(9, LOW);
    digitalWrite(7, LOW);
    digitalWrite(6, LOW);
  }
}
```