



**PROJECT REPORT**

**MEMORY GAME**

**COMPUTER ORGANIZATOIN & ASSEMBLY LANGUAGE**

**INTRODUCTION**

<b><u>NAME</u></b>	ABDUL HANNAN
<b><u>SAPID</u></b>	39115
<b><u>DEPARTMENT</u></b>	COMPUTER SCIENCE
<b><u>SESSION</u></b>	SUMMER-2024
<b><u>SUBMIT TO</u></b>	MAM DUA

## Memory Game with Blinking Lights :

In this project, a sequence of coloured lights will blink, and the user must repeat the sequence by pressing corresponding buttons. If the user enters the sequence correctly, the game will extend the sequence by one light and proceed to the next round. If the user makes a mistake, the game will end and display their score, which corresponds to how many rounds they completed successfully.

### Project Components

1. **Microcontroller:** You will need a microcontroller such as Arduino, PIC, or any other compatible microcontroller to handle the logic of the game, control the lights, and register user inputs.
2. **LEDs:** Choose LEDs in different colours (e.g., Red, Green, Blue, Yellow) or RGB LEDs to represent each possible colour in the sequence.
3. **Push Buttons:** You will need one push button for each colour. The user will press these buttons to enter the sequence they have observed.
4. **Resistors:** You will need appropriate resistors for the LEDs to ensure they operate safely.
5. **Power Supply:** The microcontroller and LEDs will need a power source, typically via USB for Arduino or a battery pack.

### Game Logic Flow

1. **Initialization:**
  - Set up the microcontroller, LEDs, and buttons.
  - Initialize variables for tracking the sequence, user input, and current score.
2. **Random Sequence Generation:**
  - Start the game by generating a random sequence of colours (represented by the LEDs).
  - Begin with one light and increase the length of the sequence by one each round.
3. **Display Sequence:**
  - Blink the LEDs in the generated sequence. For example, if the sequence is Red, Green, Blue, the LEDs will blink in that order.
  - Give the user a brief pause between the blinks for visibility.
4. **User Input:**
  - After displaying the sequence, prompt the user to replicate the sequence using the push buttons.
  - Compare the user's input to the generated sequence.
5. **Check Input:**
  - If the user inputs the correct sequence, extend the sequence by one light and repeat the process for the next round.
  - If the user makes an incorrect input, the game ends.
6. **End Game and Display Score:**
  - When the user inputs the wrong sequence, stop the game and display the score, which is the number of rounds they successfully completed.
7. **Reset:**
  - Allow the user to reset the game and start a new session.