PROJECT NAME : Number Guessing Game

Student Name: Hitesh patial **UID: 24bca10449**

Branch: BCA **Section/Group:7-A**

Semester: 1 **Date of Performance:**

Subject Name: Computer programing **Subject Code: 24CAH101**

1. **Aim/Overview of the practical:**

To develop **Number Guessing Game** in the C programming language.

1. **Task to be done:**

. Set Up the Development Environment

. Write the Program

. Add Input Validation (Optional)

.Test the Program

.Document the Code

.Reflect and Improve

1. **Algorithm/Flowchart :**

1. **Start**

2. **Initialize Random Number Generator**:

- Use `srand(time(0))` to seed the random number generator.

- Generate a random number between 1 and 100 using `rand() % 100 + 1`.

3. **Initialize Variables**:

- Set `guess` to 0 (user’s guess).

- Set `attempts` to 0 (counter for number of attempts).

4. **Prompt the User**:

   - Display: "Guess a number between 1 and 100".

5. **Input User Guess**:

   - Read user’s input and store it in `guess`.

6. **Compare the Guess with the Random Number**:

   - If `guess > random number`, display "Too high!" and go to Step 5.

   - If `guess < random number`, display "Too low!" and go to Step 5.

   - If `guess == random number`, go to Step 7.

7. **Correct Guess**:

   - Display "Congratulations! You guessed the number".

   - Display the number of attempts made.

   - End the game.

8. **Increment Attempts Counter**:

   - Increment `attempts` by 1 after each guess.

   - Repeat from Step 5 until the correct number is guessed.

9. **End**

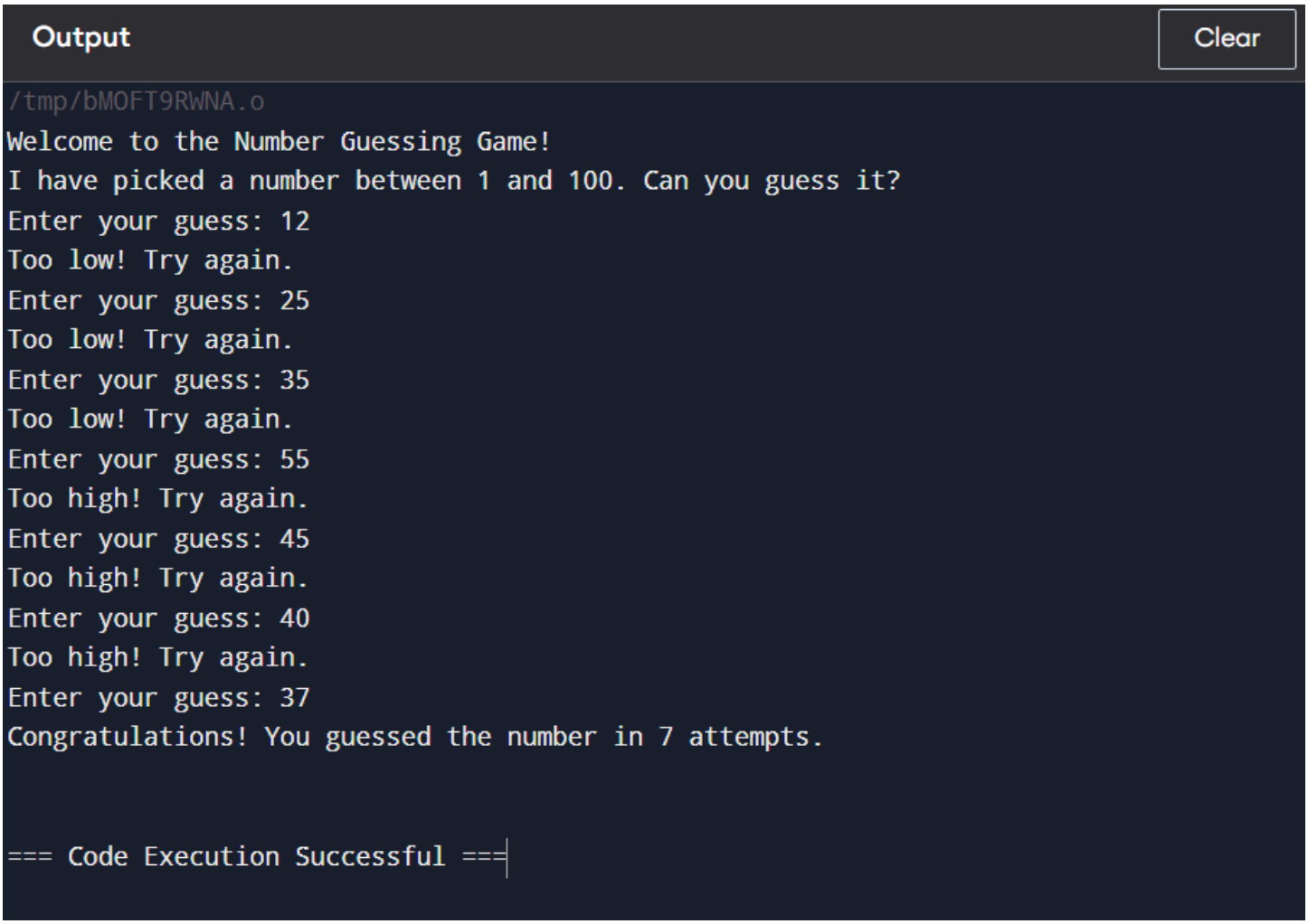
. FLOWCHART :

1. **Code for experiment/practical:**

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main() {
    int number, guess, attempts = 0;

    // Seed the random number generator with the current time
```

```
    srand(time(0));
    number = rand() % 100 + 1;  // Generate a random number between 1 and 100

    printf("Welcome to the Number Guessing Game!\n");
    printf("I have picked a number between 1 and 100. Can you guess it?\n");

    // Infinite loop until the correct guess is made
    while (1) {
        printf("Enter your guess: ");
        scanf("%d", &guess);

        attempts++;

        if (guess > number) {
            printf("Too high! Try again.\n");
        } else if (guess < number) {
            printf("Too low! Try again.\n");
        } else {
            printf("Congratulations! You guessed the number in %d attempts.\n", attempts);
            break;  // Exit the loop when the correct number is guessed
        }
    }

    return 0;
}
```

1. **Result/Output/Writing Summary:**



Writing Summary:

In this practical, we implemented a simple number-guessing game in C. The program begins by generating a random number between 1 and 100. The player is prompted to guess the number, and feedback is provided based on whether their guess is too high, too low, or correct. The program keeps track of the number of attempts and displays the result once the user guesses correctly.

This exercise demonstrated the use of control structures, user input handling, and random number generation in C programming. Additionally, the program offers an interactive experience, reinforcing key programming concepts such as loops, conditionals, and input/output functions.

Learning outcomes (What I have learnt):

1.Understanding Random Number Generation

2.Developing Game Logic

3.TrackIng and Counting Attempts

4.Improving Debugging and Testing Skills

5.Implementing Feedback Mechanisms

6.Implementing User Input Handling

7.Problem Solving and Logical Thinking

Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Pre Lab Quiz)		5
2.	Worksheet		10
3.	Post Lab Quiz		5