UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Number Guessing game

Student Name: PARAS JAIN **UID:24BCA10454**
Branch: BCA **Section/Group: 24BCA7(A)**
Semester:1 **Date of Performance:24/10/2024**
Subject Name: Computer Programing **Subject Code:24CAH101**

1. **Aim/Overview of the practical:** To develop **Number Guessing Game** in the C programming language.

2. **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

3. **Algorithm/Flowchart :**

Algorithm:

- **Start**

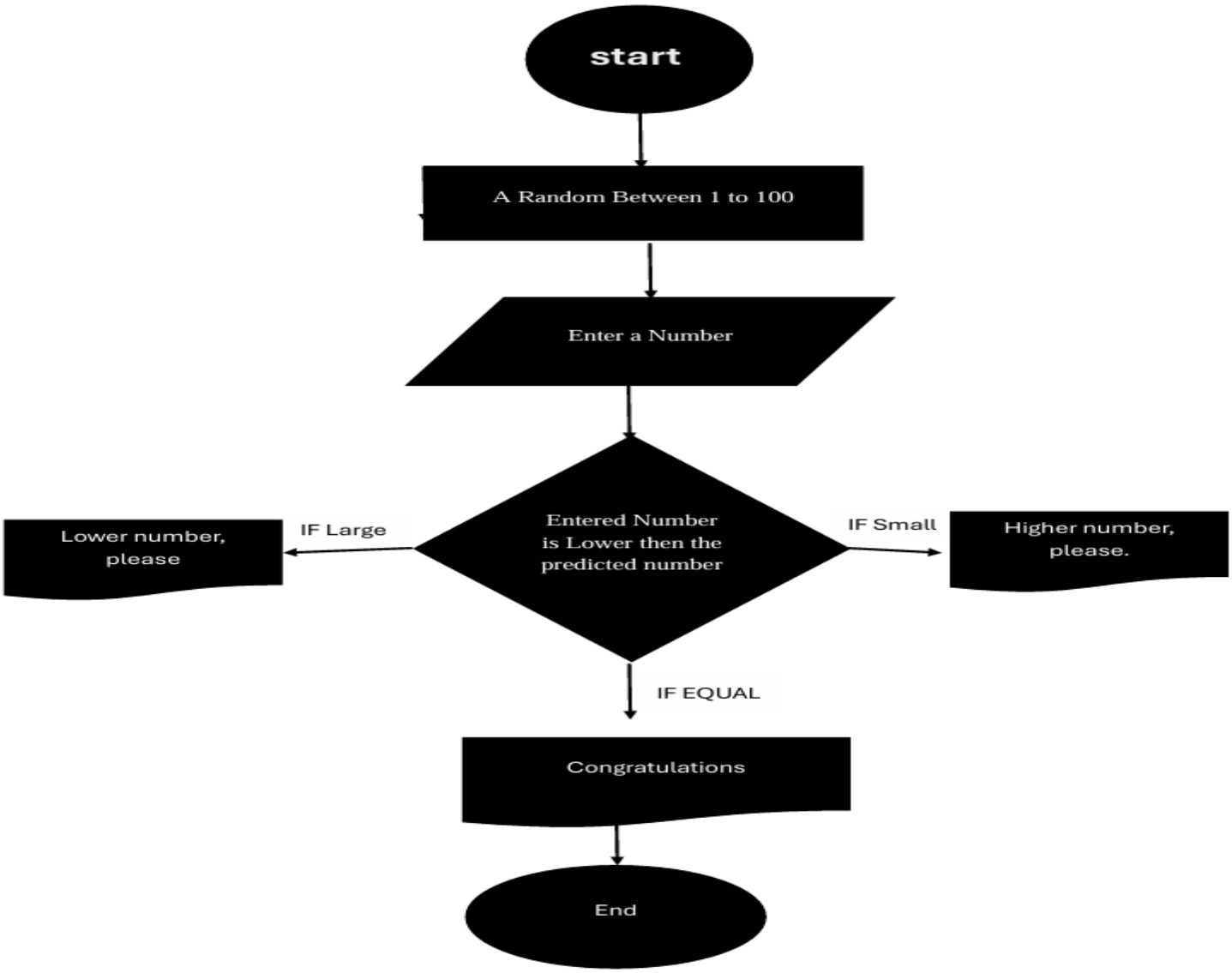
- Initialize random number generator using `srand(time(0))`.
- Generate a random number between 1 and 100.
- Set `num` (number of guesses) to 0.
- Enter a loop that will continue until the correct guess is made.

    1. Prompt the user to guess the number.
    2. Read the user's guess.
    3. Compare the guessed number with the generated random number:
        - If the guess is greater than the random number, display: "Lower number, please."
        - If the guess is less than the random number, display: "Higher number, please."
        - If the guess matches the random number, display: "Congratulations!!"
    4. Increment the guess counter (`num`).

- Exit the loop once the correct guess is made.

- Display the total number of guesses taken.
- **End**

Flowchart:



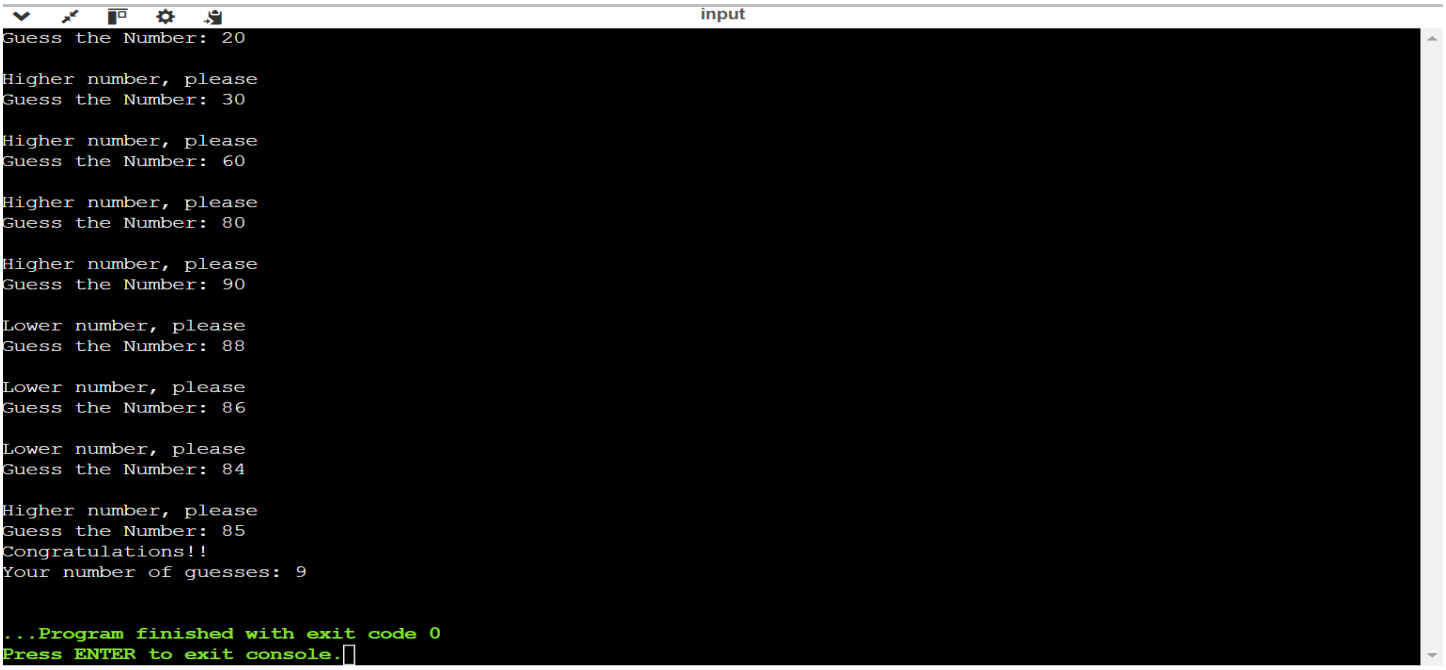
4. Code for experiment/practical:

```
#include <stdio.h>

#include <stdlib.h>

#include <time.h>
```

```
void main() {
   // Seed the random number generator with the current time
   srand(time(0));

   // Generate a random number between 1 and 100
   int randomNumber = (rand() % 100) + 1;
   int guess;  // Variable to store the user's guess
   int num = 0;  // Counter to track the number of guesses

   // Loop until the user guesses the correct number
   do {
      // Prompt the user to enter a guess
      printf("Guess the Number: ");
      scanf("%d", &guess);  // Read the user's guess

      // Compare the guess with the random number
      if (guess > randomNumber) {
         // If the guess is higher than the random number, give feedback
         printf("\nLower number, please\n");
      } else if (guess < randomNumber) {
         // If the guess is lower than the random number, give feedback
         printf("\nHigher number, please\n");
      } else {
         // If the guess is correct, congratulate the user
         printf("Congratulations!!\n");
      }
```

```
 num++;  // Increment the guess counter

  } while (guess != randomNumber);  // Continue the loop until the correct guess is made


  // After the loop ends, display the total number of guesses

  printf("Your number of guesses: %d\n", num);

}}
```

5. Result/Output/Writing Summary:



Writing Summary:

In this project, I analyzed and implemented a simple C program for a number-guessing game. The program generates a random number between 1 and 100 and allows the user to make guesses. Based on the user's input, the program provides feedback on whether the guess is too high or too low, and it continues until the correct number is guessed. After the correct guess, the program outputs the total number of attempts. Additionally, I designed an algorithm and flowchart to visualize the steps involved in the game.

Learning outcomes (What I have learnt):

1. Learned how to use the `rand()` function in C for random number generation.
2. Understood the use of loops to handle repeated user inputs.
3. Gained experience in using `if-else` conditions to provide feedback based on comparisons.
4. Improved skills in tracking user attempts and creating clear visual flowcharts.
5. Developed the ability to translate program logic into an algorithm.

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