

# PROJECT REPORT

## NUMBER GUESSING GAME USING C

---

**Project Title:** Number Guessing Game in C

**Course/Subject:** Programming in C

**Prepared By:** Rakshit Jangid

**Roll No:** 590028078

**Submitted To:** Srinivasan Ramachandran

**Batch:** 62

---

## 2. Abstract

This project presents a simple yet interactive *Number Guessing Game* developed in the C programming language.

The program generates a random number using the `rand()` function and asks the user to guess it. Based on the input, the program provides hints such as “*low*” or “*high*” until the correct number is guessed.

This project demonstrates core programming concepts including loops, conditional statements, user input handling, and random number generation. The system is lightweight, easy to execute, and suitable for beginners learning logic building in C.

---

## 3. Problem Definition

The objective is to design a simple interactive program where:

1. The computer randomly selects a number between **1 and 10**.
2. The user must guess the number.
3. After each guess, the program tells the user whether the guess is:
  - **Correct**
  - **Too low**
  - **Too high**
4. The program continues running until the user guesses correctly.

The purpose is to practice and demonstrate:

- Basic C syntax
- Random number generation
- Loops

- Decision-making
  - Input/output handling
- 

## 4. System Design

### 4.1 Flowchart

---

### 4.2 Algorithm

**Step 1:** Start the program

**Step 2:** Declare variables num and guess

**Step 3:** Initialize random generator using srand(time(0))

**Step 4:** Generate random number num = rand() % 10 + 1

**Step 5:** Display "Guess the number (1 to 10)"

**Step 6:** Repeat forever

a. Take input from user → guess

b. If guess == num

    → Print "You got it!!"

    → Break the loop

c. Else if guess < num → Print "low.. try again"

d. Else → Print "high.. try again"

**Step 7:** End program

---

## 5. Implementation Details

### 5.1 Code Snippet (Provided Program)

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

int main() {
    int num, guess;

    srand(time(0)); // random
    num = rand() % 10 + 1;
    printf("Guess the number (1 to 10): ");

    while(1) {
        scanf("%d", &guess);

        if(guess == num){
            printf("You got it!!\n");
            break;
        }
    }
}
```

```

    }
    else if(guess < num){
        printf("low.. try again: ");
    }
    else{
        printf("high.. try again: ");
    }
}

return 0;
}

```

## 5.2 Key Features Used

- **Header Files**
    - `<stdio.h>` – input/output
    - `<stdlib.h>` – random functions
    - `<time.h>` – seeding random generator
  - **Random Number Generation**
    - `srand(time(0))` ensures different number each run
    - `rand() % 10 + 1` generates a number from 1 to 10
  - **Infinite Loop**
    - `while(1)` keeps program running until correct guess
  - **Decision Making**
    - `if-else` statements compare user guess
- 

# 6. Testing & Results

## 6.1 Test Case Table

Test Case	Input Guess	Expected Output	Actual Output	Result
1	Correct number	"You got it!!"	Same	Pass
2	Lower than number	"low.. try again"	Same	Pass
3	Higher than number	"high.. try again"	Same	Pass
4	Multiple wrong guesses	Repeats hints	Same	Pass
5	Correct guess after multiple tries	Ends loop	Same	Pass

## 6.2 Sample Run

```

Guess the number (1 to 10): 3
low.. try again: 7
high.. try again: 5
You got it!!

```

The program behaves exactly as intended.

---

## 7. Conclusion & Future Work

### Conclusion

The Number Guessing Game successfully demonstrates the basic concepts of C programming including loops, conditionals, and random number generation. The project is simple, interactive, and effective for learning logic development.

### Future Enhancements

- Add limited attempts (e.g., only 5 tries).
  - Add difficulty levels (Easy, Medium, Hard).
  - Expand range (1–100 or 1–1000).
  - Display number of attempts taken by the user.
  - Create a menu-driven version.
  - Store scores or high scores in a file.
- 

## 8. References

- *Let Us C* – Yashavant Kanetkar
  - C Programming Language (ANSI C) documentation
  - TutorialsPoint – C Programming Basics
  - GeeksforGeeks – Random Number Generation in C
-