UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Snake,Water and Gun game

Student Name: Piyush Jana **UID:24BCA10472**
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 **Snake,Water and Gun 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:

1. **Start**
2. **Initialize Random Number Generator:**
   Use `srand(time(0))` to ensure that the computer generates a random choice in each round.
3. **Prompt User Input:**

Display the options to the user:

0 for Snake

1 for Water

2 for Gun

Accept and store the user's choice.

4. **Validate User Input:**

Check if the input is valid (0, 1, or 2).

If the input is invalid (not in the range), display an error message "Invalid choice!" and terminate the program.

5. **Computer's Choice:**

Use `rand() % 3` to randomly generate the computer's choice (0 for Snake, 1 for Water, 2 for Gun).

6. **Display the Computer's Choice:**

Show the corresponding choice (Snake, Water, or Gun) for the computer.

7. **Determine the Winner:**

If the user and computer made the same choice, display "It's a tie!"

Otherwise, apply the game rules to determine the winner:

Snake beats Water.

Water beats Gun.

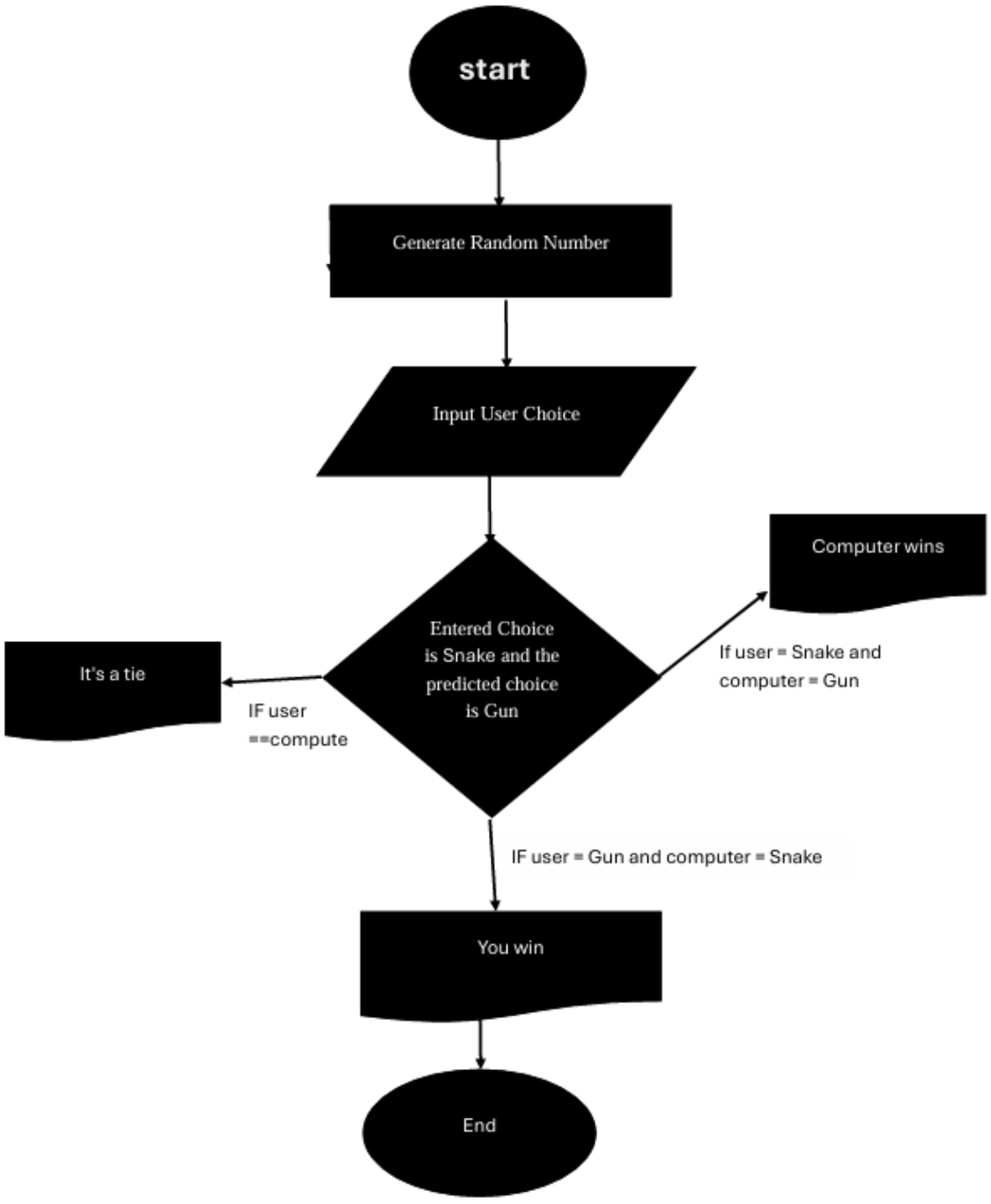
Gun beats Snake.

If the user's choice wins based on these rules, display "You win!"

If the computer's choice wins, display "Computer wins!"

8. **End**

Flowchart:



4.Code for experiment/practical:

```
#include <stdio.h>

#include <stdlib.h>
```

```
#include <time.h>

void display(int choice) {
    switch (choice) {
        case 0: printf("Snake\n"); break;
        case 1: printf("Water\n"); break;
        case 2: printf("Gun\n"); break;
        default: printf("Invalid choice\n");
    }
}

int main() {
    int user, computer;

    srand(time(0)); // Seed the random number generator

    // User input for their choice
    printf("Enter your choice (0: Snake, 1: Water, 2: Gun): ");
    scanf("%d", &user);

    // Validate user input
    if (user < 0 || user > 2) {
        printf("Invalid choice! Please try again.\n");
        return 1;  // Exit the program if input is invalid
```

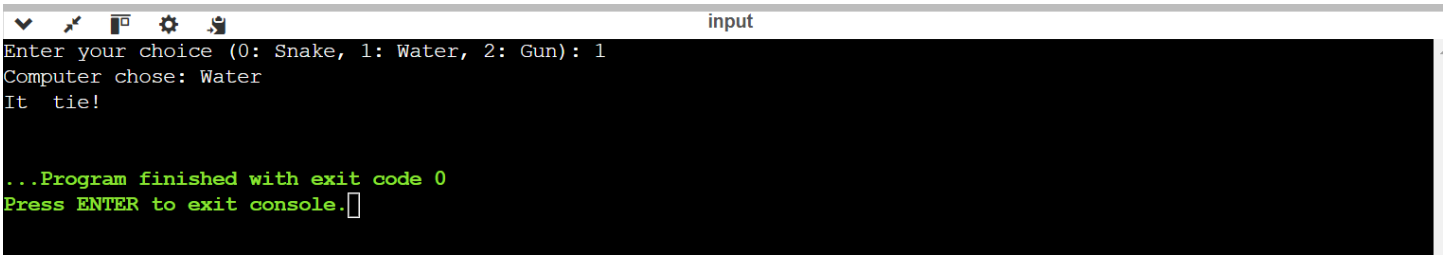
```
    }

    // Computer makes a random choice
    computer = rand() % 3;
    printf("Computer chose: ");
    display(computer);

    // Determine the winner
    if (user == computer) {
        printf("It  tie!\n");
    } else if ((user == 0 && computer == 1) || // Snake drinks water
               (user == 1 && computer == 2) || // Water douses gun
               (user == 2 && computer == 0)) { // Gun kills snake
        printf("You win Congratulations!\n");
    } else {
        printf("Computer wins!\n");
    }

    return 0;
}
```

5.Result/Output/Writing Summary:



Writing Summary:

This C program implements the "Snake, Water, Gun" game where the user competes against the computer. The user selects Snake, Water, or Gun, and the computer makes a random choice. The outcome is determined by game rules: Snake beats Water, Water beats Gun, and Gun beats Snake. The game ends after one round, displaying the result (win, lose, or tie). The project highlights user input, random number generation, and conditional logic for determining the winner.

6.Learning outcomes (What I have learnt):

1.User Input Handling: Learned how to gather and validate user input in C, ensuring the program handles both valid and invalid choices effectively.

2.Random Number Generation: Gained experience using `rand()` and `srand()` to implement random computer choices, a key feature in game development.

3.Conditional Logic Application: Improved my skills in using `if-else` statements to compare user and computer choices and determine the game outcome.

4.Game Development: Understood the process of designing a simple interactive game, focusing on user experience and clear feedback.

5.Efficient Code Structuring: Practiced writing modular and readable code, separating the logic of displaying choices into functions for better organization.

6.Debugging and Problem Solving: Enhanced my ability to debug, ensuring the game functions correctly for various inputs and scenarios.

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