KALASALINGAM
ACADEMY OF RESEARCH & EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade
Anand Nagar, Krishnankoil - 626126. Srivilliputtur (Via), Virudhunagar (Dt), Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

PROJECT REPORT
ROCK PAPER AND SCISSOR

Section name : B02

Group Number :

Group Leader : P. Linu Zenas Paul

Members of the Team : 02

S.NO	NAME OF THE STUDENT	REGISTER NUMBER	SIGNATURE OF THE STUDENT
1	K. Veera Gopi	9923005171	
2	P. Linu Zenas Paul	9923008040	

Department of Computer Science and Engineering

School of Freshman Engineering

Kalasalingam Academy of Research and Education

Signature of the Examiner with Date

I. INTRODUCTION

The project implements a classic Rock, Paper, Scissor game where users can play against the computer. The program welcomes users, prompts them to choose a move (Rock, Paper, or Scissor), generates a random move for the computer, compares the choices, determine the winner, updates and displays scores, and provides an option to play again and quit.

The game incorporates essential C concepts such as input, random number generation, conditional statements, loops, and scorekeeping. It aims to create an interactive and enjoyable experience while demonstrating fundamental programming skills. The simplicity of the project makes it suitable for beginners, allowing them to practice and understand basic programming constructs in the context of a fun and timeless game.

The code structure is organized, and the project encourage further enhancements, such as improving user interface, handling invalid inputs, or extending the game with additional feature. It serves as a practical exercise for those learning C programming, offering a hands-on approach to implementing logic and control flow in a real-world application.

This Rock, Paper, Scissor project provides an engaging introduction to programming logic and user interaction in C. The random computer move adds an element of unpredictability, challenging players to strategize and make decisions quickly. The scoring system keeps track of both user and computer performance.

As users play the game, they gain exposure to foundational programming concepts like functions, conditional statements, and loops. The project's concise design ensures that learners can grasp the core mechanics of the game while providing a foundation for exploring more complex programming challenges in the future. The modular structure of the code allows for easy expansion and improvement. Beginners can experiment with additional features, such as creating a graphical user interface or implementing a history log of previous moves. Furthermore, error handling mechanisms can be incorporated to enhance the user experience by gracefully managing unexpected inputs.

II. OBJECTIVE OF THE PROJECT

The primary objective of this Rock, Paper, Scissors project in C is to create an interactive and educational application that introduces fundamental programming concepts to beginners. The project aims to achieve the following goals:

1. **Introduction to C Programming**:

- Provide a hands-on experience for beginners to practice programming in C.
- Familiarize users with basic syntax, user input, conditional statements, and loops.

2. **User Interaction and Decision Making**:

- Implement a user-friendly interface that allows players to make choices interactively.
- Utilize conditional statements to compare user and computer moves, determining winner.

3. **Randomization and Game Logic**:

- Introduce the concept of randomness by generating a random move for the computer.
- Implement the core logic of the Rock, Paper, Scissors game, incorporating conditional checks for winning conditions.

4. **Scoring System**:

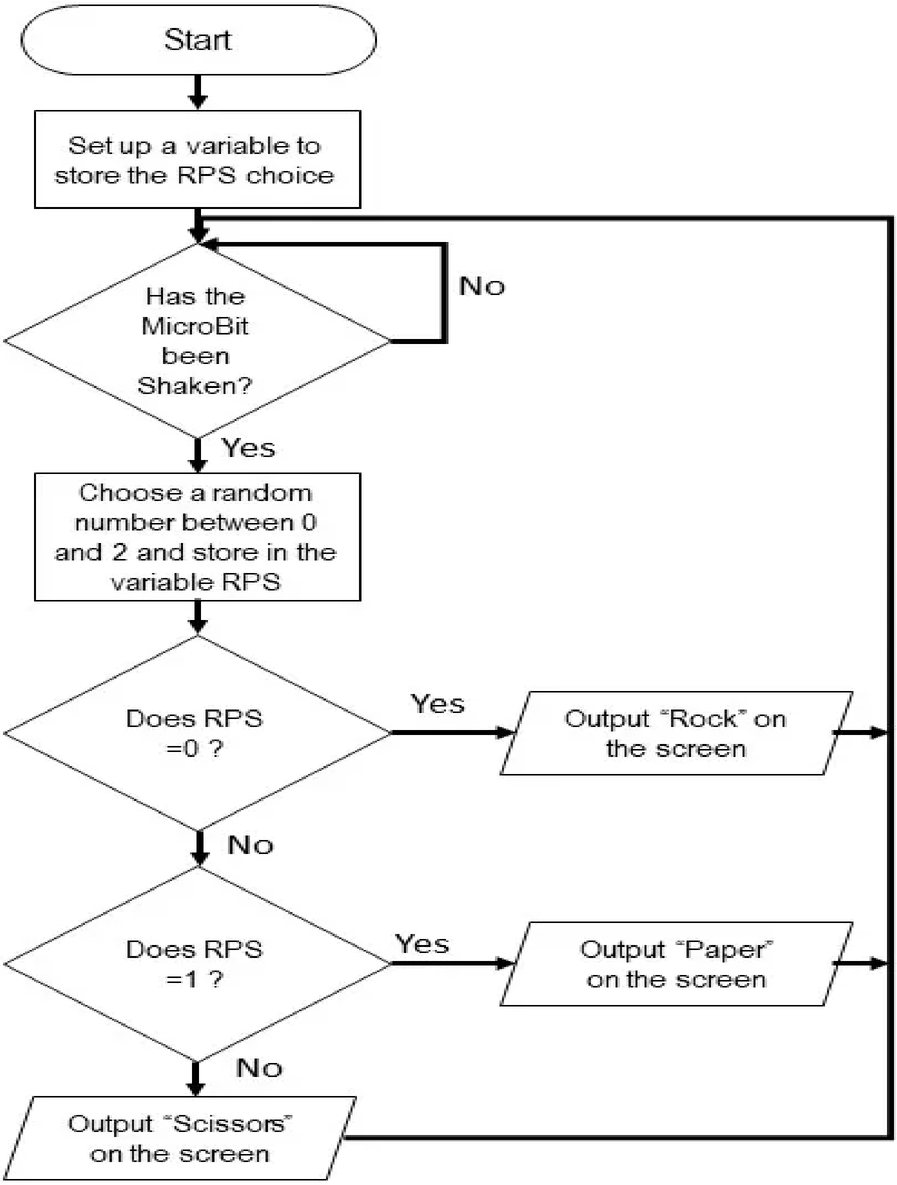
- Develop a scoring system to keep track of both user and computer performance throughout the game.
- Reinforce arithmetic operations by updating and displaying scores based on game output.

5. **Modularity and Expandability**:

- Design the code in a modular manner, allowing for easy understanding and potential expansion.
- Encourage users to experiment with enhancements, such as error handling, graphical interfaces, or additional game features.

III. DESIGN OF THE SYSTEM

FLOWCHART


Start
Set up a variable to store the RPS choice
Has the MicroBit been Shaken?
No
Yes
Choose a random number between 0 and 2 and store in the variable RPS
Does RPS =0 ?
Yes
Output "Rock" on the screen
No
Does RPS =1 ?
Yes
Output "Paper" on the screen
No
Output "Scissors" on the screen

IV. SYSTEM SPECIFICATION / PROJECT MODULES DESCRIPTION

1.User input Module:

- Get user input for their choice(rock, paper or scissor).
- Validate the input to ensure it's a valid choice.

2.Computer Choice Module:

- Generate a rondom choice for the computer (rock, paper or scissor).

3.Game Logic Module:

- Compare the user's choice with computer's choice.
- Determine the winner or it it's a tie.
- Update the score accordingly.

4.Score Tracking Module:

- Keep track of the user's and computer's score.
- Create variables to store the scores of the players. For example, you might have variables like player1Score and player2Score. Initialize them to zero at the beginning of the game.

5.Display Module:

- Show the choice made by the use and the computer.
- Display the winner or it's a tie.
- Show the current score.

6.Main Game Loop:

- Run the game in a loop until the user decides to quit.
- Allow the user to play multiple rounds.

7.Exit Module:

- Within the exit module, you might print a farewell message, display the final score, or perform any necessary cleanup actions
- Provide an option for the user to exit the game.

V. IMPLEMENTATION

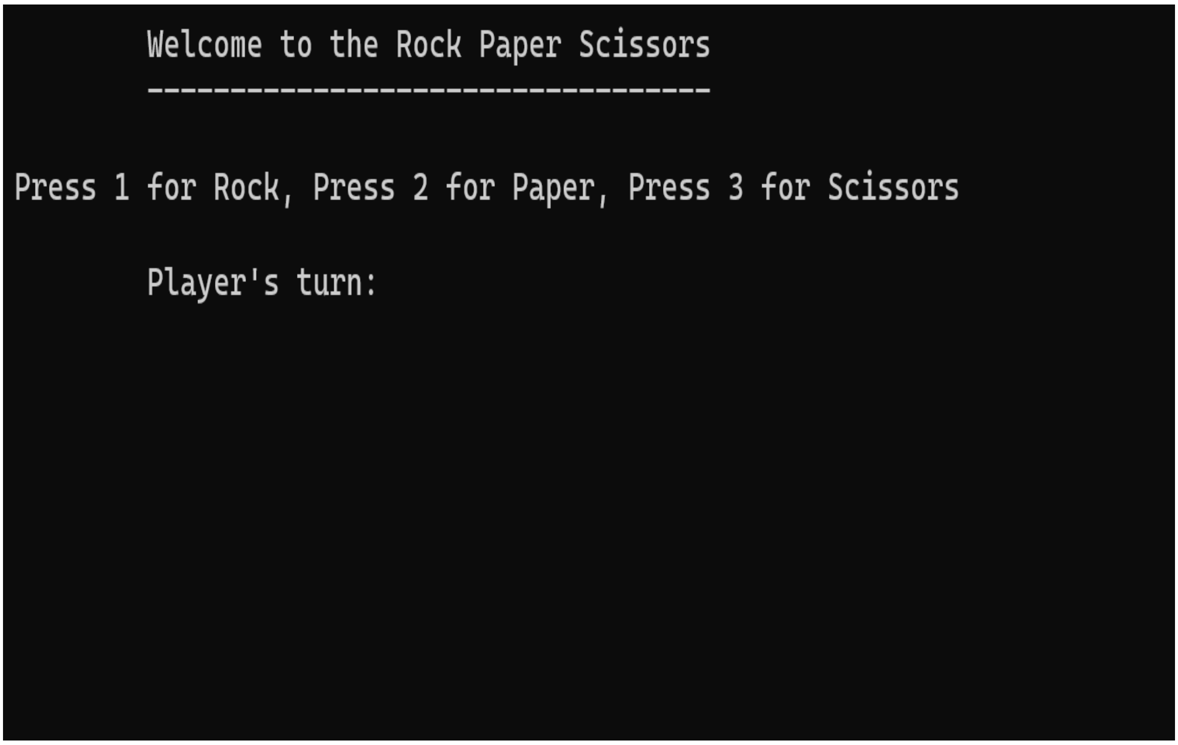
```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int generateRandomNumber(int n)
{
srand(time(NULL));
return rand() % n;
}

int greater(char c1, char c2)
{
if (c1 == c2)
{
return -1;
}
else if (c1 == 'r' && c2 == 's')
{
return 1;
}
else if (c2 == 'r' && c1 == 's')
{
return 0;
}
else if (c1 == 'p' && c2 == 'r')
{
return 1;
}
else if (c2 == 'p' && c1 == 'r')
{
return 0;
}

else if (c1 == 's' && c2 == 'p')
{
return 1;
}
else if (c2 == 's' && c1 == 'p')
{
return 0;
}
}
int main()
{
int playerScore = 0, compScore = 0, temp;
char playerChar, compChar;
char dict[] = {'r', 'p', 's'};
printf("\tWelcome to the Rock Paper Scissors\n");
printf("\t-----------------------------------\n\n");

for (int i = 0; i < 3; i++)
```

```
{
printf("Press 1 for Rock, Press 2 for Paper, Press 3 for Scissors\n\n");
printf("\tPlayer's turn: ");

scanf("%d", &temp);
getchar();
playerChar = dict[temp - 1];
printf(" -----------------\n");
printf("| You choose: %c   |\n", playerChar);
printf(" -----------------\n\n");

printf("Press 1 for Rock, Press 2 for Paper, Press 3 for Scissors\n\n");
printf("\tComputer's turn\n");
temp = generateRandomNumber(3) + 1;
compChar = dict[temp - 1];
printf(" --------------------\n");
printf("| Computer choose: %c |\n", compChar);
printf(" --------------------\n\n");

if (greater(compChar, playerChar) == 1)
{
compScore++;
printf("\t\tComputer Got It!\n\n");
}
else if (greater(compChar, playerChar) == -1)
{
compScore++;
playerScore++;
printf("\t\tIt's a draw. Both got 1 point!\n\n");
}
else
{
playerScore++;
printf("\t\tYou Got It!\n\n");
}

printf(" -------------\n");
printf("| You: %d      |\n", playerScore);
printf("| Computer: %d |\n", compScore);
printf(" -------------\n\n");

printf("==========================================================\n\n");
}

printf(" -----------------\n");
printf("|   Final Score   |\n");
printf(" -----------------\n");
printf("|  You | Computer |\n");
printf("|------|----------|\n");
printf("|   %d  |    %d     |\n", playerScore, compScore);
printf(" -----------------\n\n");

if (playerScore > compScore)
{
```

```
printf("\n\t -------------------\n");
printf("\t| You Win the match |\n");
printf("\t -------------------\n");
}
else if (playerScore < compScore)
{
printf("\n\t ------------------------\n");

printf("\t| Computer Win the match |\n");
printf("\t ------------------------\n");
}
else
{
printf("\n\t -------------\n");
printf("\t| It's a draw |\n");
printf("\t -------------\n");
}

return 0;
}
```

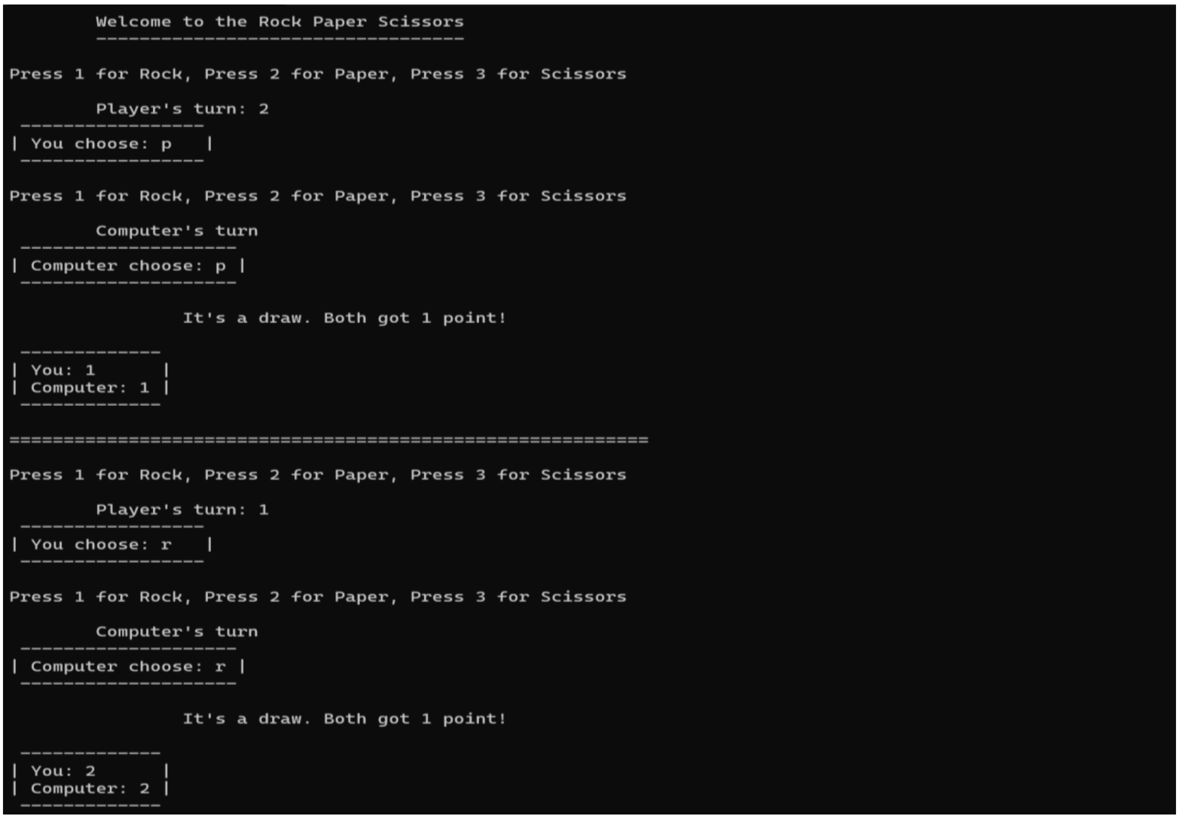
VI. IMPLEMENTATION SCREENSHOTS

1. WELCOME



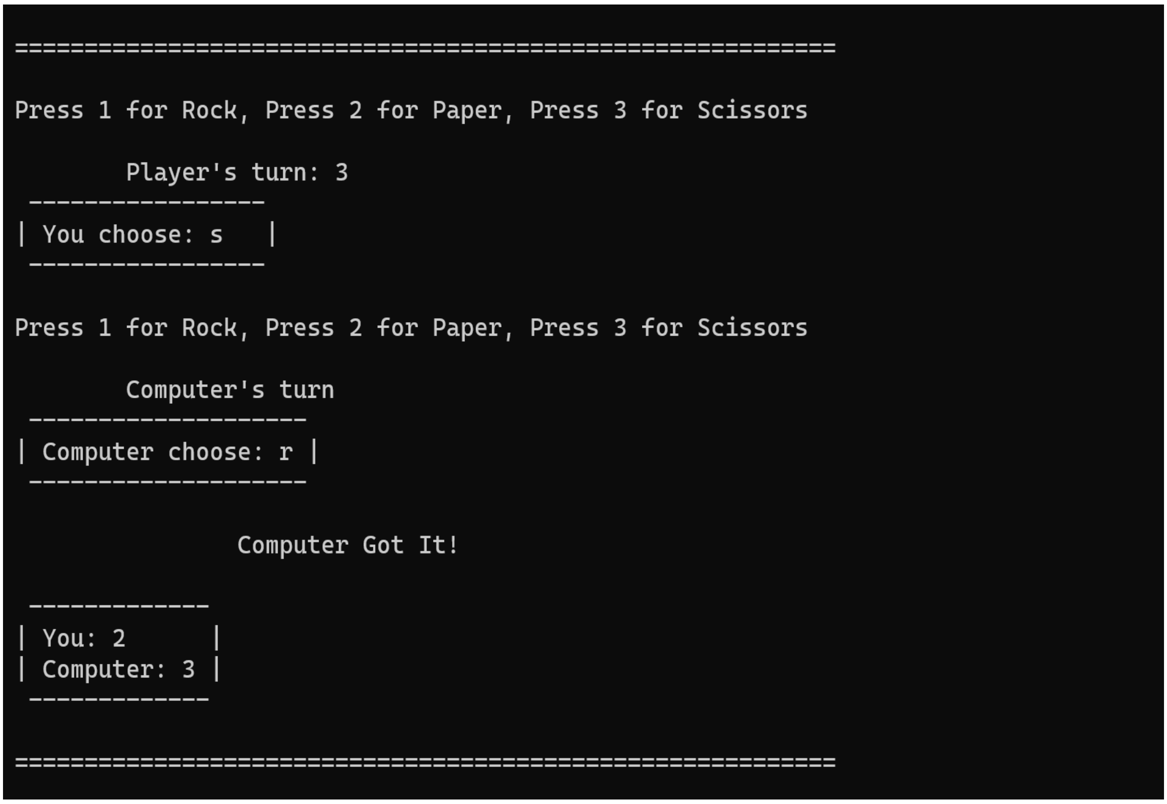
```
        Welcome to the Rock Paper Scissors
        ----------------------------------

Press 1 for Rock, Press 2 for Paper, Press 3 for Scissors

        Player's turn:
```

2. PLAYING



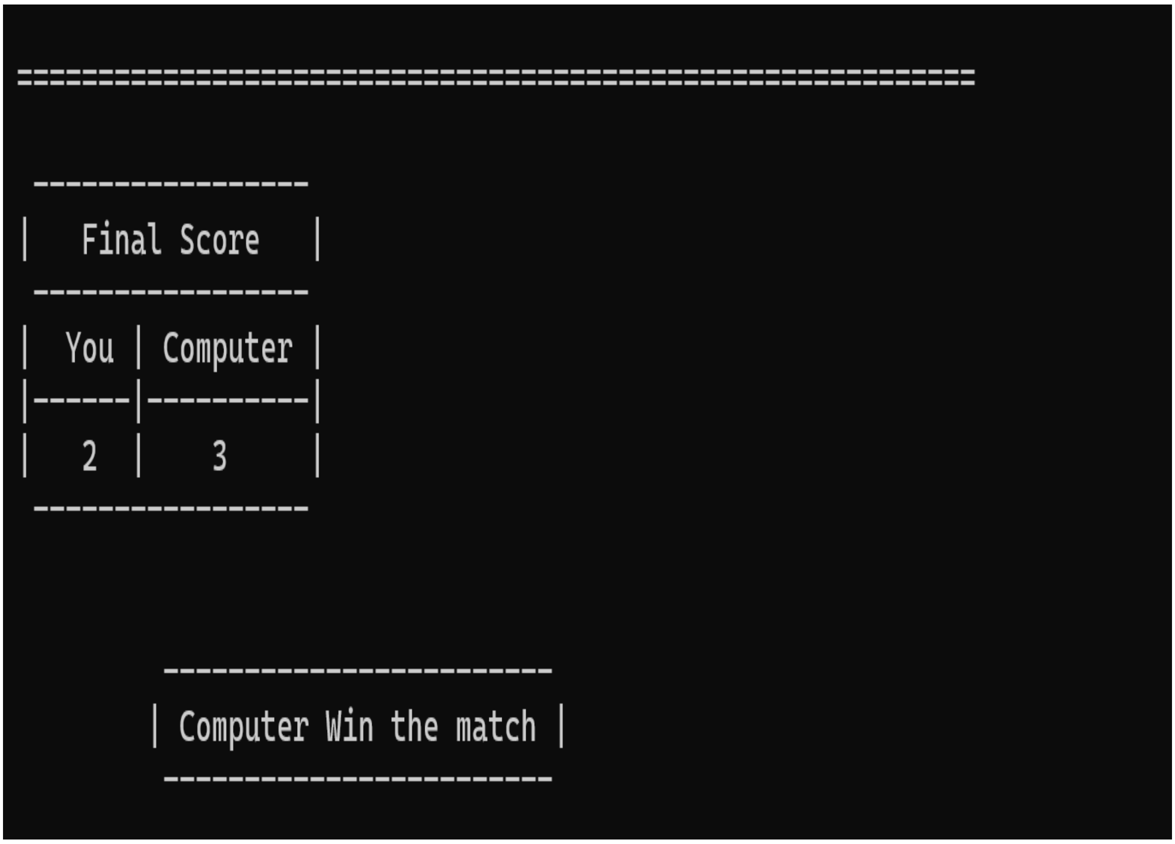
```
        Welcome to the Rock Paper Scissors
        ----------------------------------

Press 1 for Rock, Press 2 for Paper, Press 3 for Scissors

        Player's turn: 2
 -----------------
| You choose: p   |
 -----------------

Press 1 for Rock, Press 2 for Paper, Press 3 for Scissors

        Computer's turn
 --------------------
| Computer choose: p |
 --------------------

                It's a draw. Both got 1 point!

 -------------
| You: 1      |
| Computer: 1 |
 -------------

===========================================================

Press 1 for Rock, Press 2 for Paper, Press 3 for Scissors

        Player's turn: 1
 -----------------
| You choose: r   |
 -----------------

Press 1 for Rock, Press 2 for Paper, Press 3 for Scissors

        Computer's turn
 --------------------
| Computer choose: r |
 --------------------

                It's a draw. Both got 1 point!

 -------------
| You: 2      |
| Computer: 2 |
 -------------
```

3. PLAYING



```
===========================================================

Press 1 for Rock, Press 2 for Paper, Press 3 for Scissors

        Player's turn: 3
 -----------------
| You choose: s   |
 -----------------

Press 1 for Rock, Press 2 for Paper, Press 3 for Scissors

        Computer's turn
 --------------------
| Computer choose: r |
 --------------------

                Computer Got It!

 -------------
| You: 2      |
| Computer: 3 |
 -------------

===========================================================
```

4. FINAL SCORE



```
============================================================


 -----------------
|   Final Score   |
 -----------------
|  You | Computer |
|------|----------|
|   2  |    3     |
 -----------------



         ------------------------
        | Computer Win the match |
         ------------------------
```

VII. CONCLUSION AND FUTURE ENHANCEMENT

In conclusion, this Rock, Paper, Scissors project in C has successfully demonstrated the implementation of a classic game, providing an interactive and educational programming experience. The project's concise structure introduces fundamental concepts such as user input, randomization, and conditional logic, making it an excellent resource for beginners. The modular design encourages future enhancement and creativity. Learners can further refine the project, adding features like error handling or graphical interfaces. Overall, this project serves as an engaging foundation for individuals to build upon as they progress in their C programming journey.

This Rock, Paper, Scissors project not only introduces the fun of a classic game but also serves as a practical and enjoyable platform for individuals eager to strengthen their programming skills in C. The simplicity of the game design allows for creativity and customization, making it an ideal starting point for those on their programming journey.