

# ROCK PAPER SCISSOR

**Student Name:** Kirtee Thakur

**UID:**24BCA10467

**Branch:** BCA

**Section:**7A

**Semester:**1<sup>st</sup>

**Subject code :**24CAH101

**Subject Name:** Computer Programming

**Date of Performance:**

**1.Aim/Overview of the practical:** To develop rock paper scissor using C programming language.

**2.Task to be Done :**

- **Understand Game Rules:** Clearly define the rules of the game, including player and computer input, winning combinations, and the logic behind them (rock beats scissors, scissors beat paper, paper beats rock).
- **Design User Interface:** Plan how the game will interact with the player, including displaying instructions and asking for input (rock, paper, or scissors).
- **Generate Random Computer Move:** Implement a function to generate a random choice for the computer using a random number generator.
- **Implement Player Input:** Create a system to accept and validate input from the player (rock, paper, or scissors).
- **Determine the Winner:** Write the logic to compare the player's choice with the computer's and determine the winner.
- **Display the Results:** Print the outcome of each round, including the player's and computer's choices and whether the player won, lost, or tied
- **Repeat or Exit the Game:** Add a feature that allows the player to play multiple rounds or exit the game after each round.
- **Test and Debug:** Thoroughly test the program to ensure all features work as intended and handle invalid inputs gracefully.

**3.Algorithm/Flowchart:**

**Algorithm:**

1. Start
2. Display game instructions.
3. Generate a random number (1, 2, or 3) for the computer's move:
  - **1 = Rock**
  - **2 = Paper**
  - **3 = Scissors**
4. Ask the player to input their choice:
  - **1 = Rock**

- **2 = Paper**
- **3 = Scissors**

5. Compare the player's choice with the computer's choice:

- If both choices are the same, it's a tie.
- If the player chooses Rock:
  - Rock vs Scissors: Player wins.
  - Rock vs Paper: Computer wins.
- If the player chooses Paper:
  - Paper vs Rock: Player wins.
  - Paper vs Scissors: Computer wins.
- If the player chooses scissor:
  - Scissors vs Paper: Player wins.
  - Scissors vs Rock: Computer wins.

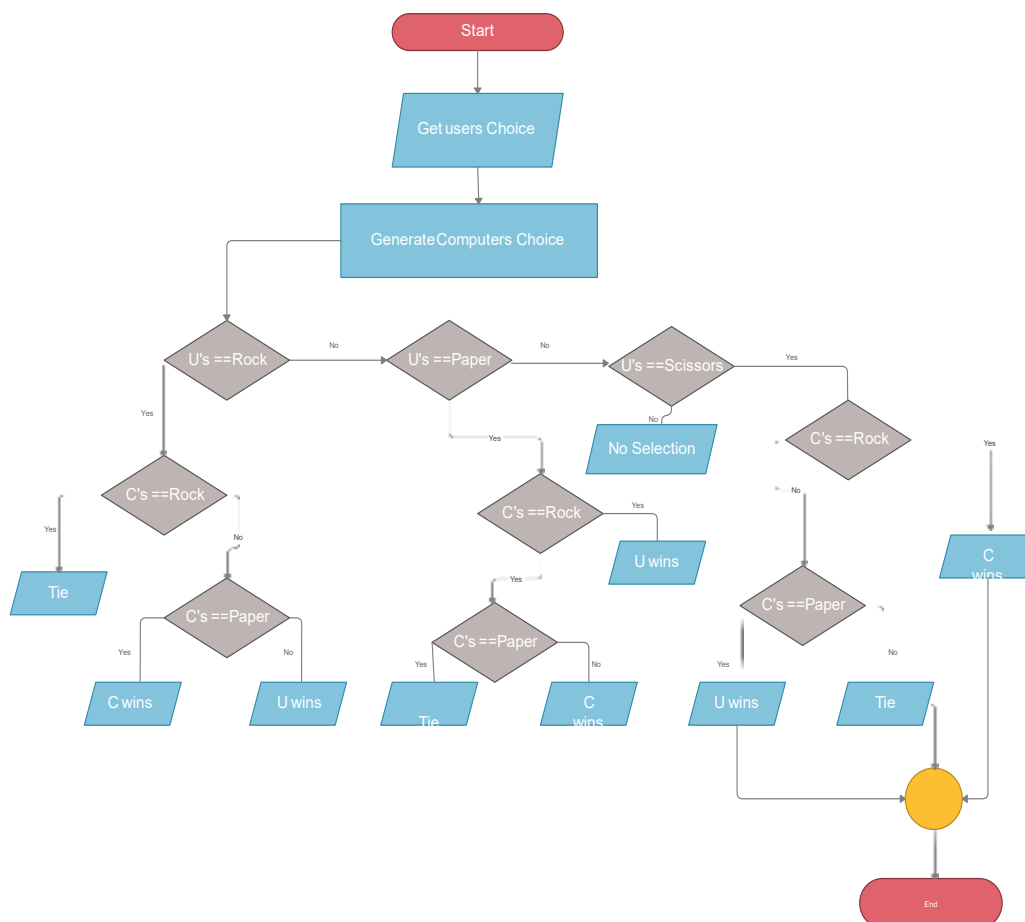
6. Display the result of the round (win, lose, or tie).

7. Ask if the player wants to play another round.

8. If yes, repeat from Step 3. If no, end the game.

9. End

## Flowchart:



#### 4.Code for Experiment :

```
#include <math.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int game(char you, char computer)
{
    if (you == computer)
        return -1;

    if (you == 's' && computer == 'p')
        return 0;

        else if (you == 'p' && computer == 's') return 1;

    if (you == 's' && computer == 'x')
        return 1;

    else if (you == 'x' && computer == 's')
        return 0;

    if (you == 'p' && computer == 'x')
        return 0;

    else if (you == 'x' && computer == 'p')
        return 1;
}

int main()
{
    int n;

    char you, computer, result;

    srand(time(NULL));

    n = rand() % 100;

    if (n < 33)
        computer = 's';

    else if (n > 33 && n < 66)
        computer = 'p';
```

```

else
    computer = 'x';
printf("\n\n\n\n\t\t\tEnter s for STONE, p for PAPER and x for
SCISSOR\n\t\t\t\t\t\t\t");
scanf("%c", &you);
result = game(you, computer);
if (result == -1) {
    printf("\n\n\t\t\t\t\tGame Draw!\n");
}
else if (result == 1) {
    printf("\n\n\t\t\t\t\tWow! You have won the game!\n");
}
else {
    printf("\n\n\t\t\t\t\tOh! You have lost the game!\n");
}
printf("\t\t\t\t\tYOU choose : %c and Computer choose : %c\n",you,
computer);

return 0;
}

```

## 5. Result /Output:

```
Output
/tmp/tSDg9J9c01.o

Enter s for STONE, p for PAPER and x for SCISSOR
x

Game Draw!
Y0u choose : x and Computer choose : x

=== Code Execution Successful ===
```

```
Output
/tmp/pceoZMDBUy.o

Enter s for STONE, p for PAPER and x for SCISSOR
s

Oh! You have lost the game!
Y0u choose : s and Computer choose : p

=== Code Execution Successful ===
```

## Output

/tmp/qBp65T7XRN.o

Enter s for STONE, p for PAPER and x for SCISSOR  
p

Oh! You have lost the game!  
Y0u choose : p and Computer choose : x

=== Code Execution Successful ===

## Output

/tmp/JMPJyV0E6d.o

Enter s for STONE, p for PAPER and x for SCISSOR  
x

Wow! You have won the game!  
Y0u choose : x and Computer choose : p

=== Code Execution Successful ===

## 6. Writing Summary:

In this project, a simple Rock-Paper-Scissors game was developed using the C programming language. The game simulates a match between a player and the computer, where both make their choices (rock, paper, or scissors), and the result is determined based on standard game rules. The program begins by explaining the game rules to the user and accepting the user's choice. The computer's choice is generated randomly. The player's and computer's selections are compared to determine the outcome of the game, which can be either a win for the player, a win for the computer, or a tie.

### Key features implemented include:

- Input validation to ensure that the player provides a valid choice.
- A random function for generating the computer's move.
- Comparison logic to evaluate the winner of each round.
- A loop that allows the player to play multiple rounds or exit the game.

## 7. Learning Outcomes:

- ✚ Understanding of Conditional Logic
- ✚ Random Number Generation
- ✚ User Input Handling:
- ✚ Looping Structures:
- ✚ Debugging and Problem-Solving
- ✚ Basic Game Design in C
- ✚ Project Management
- ✚ Code Optimization

### 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

