Rochambeau

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

/* Constants representing rock, paper, and scissors
* 0, 1, & 2 are chosen so modulo 3 can be used in 'getComputerChoice' user-defined function
*/

#define ROCK 0
#define PAPER 1
#define SCISSORS 2

/*I tried using the following code to define rock, paper and scissors as constant integers:
const int ROCK = 0;
const int PAPER = 1;
const int SCISSORS = 2;
But recieved this error from the compiler: 'error: case label does not reduce to an integer constant'
*/

// Function prototypes
int getComputerChoice();
int determineWinner(int userChoice, int computerChoice);

int main() {
    // Seeding the RNG with the current time
    srand(time(NULL));

    int numRounds;
    // Prompting for user input
    printf("Enter the number of rounds you wish to play: ");
    scanf("%d", &numRounds);

    // Declaring and assigning variables
    int userWins = 0;
    int computerWins = 0;
    int ties = 0;
```

```
//taking in user input/logic
for (int i = 0; i < numRounds; i++) {
    //declaring userInput
    char userInput;
    printf("Enter R, P, or S: ");
    scanf(" %c", &userInput);

    //switch is testing for userInput values, upper and lowercase
    //declaring userChoice
    int userChoice;
    switch (userInput) {
        case 'R': case 'r':
            userChoice = ROCK;
            break;
        case 'P': case 'p':
            userChoice = PAPER;
            break;
        case 'S': case 's':
            userChoice = SCISSORS;
            break;
        //if none of the previous conditions met, prints out error message to screen
        default:
            printf("Invalid input! Try again.\n");
            i--;// postfix decrement counter to repeat current round
            continue; // Skips the rest of the loop to start the next iteration
    }

    int computerChoice = getComputerChoice();
    int result = determineWinner(userChoice, computerChoice);
    //ties
    if (result == 0) {
        printf("Tie\n");
        ties++;
    //user wins
    } else if (result == 1) {
        printf("You won! ");
        switch (userChoice) {
            case ROCK:
                printf("Rock crushes Scissors\n");
```

```
                break;
            case PAPER:
                printf("Paper covers Rock\n");
                break;
            case SCISSORS:
                printf("Scissors cut Paper\n");
                break;
        }
        userWins++; //counting number of user wins for output statement at the end
    //computer wins
    } else {
        printf("Computer won! ");
        switch (computerChoice) {
            case ROCK:
                printf("Rock crushes Scissors\n");
                break;
            case PAPER:
                printf("Paper covers Rock\n");
                break;
            case SCISSORS:
                printf("Scissors cut Paper\n");
                break;
        }
        computerWins++; //counting number of computer wins for output statement at
the end
    }
  }
//output statement after rounds have been played
  printf("Game Over!\n");

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~\n");
  printf("You won %d rounds. The computer won %d rounds and there were %d
ties.\n", userWins, computerWins, ties);

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~\n");

  return 0;
}
```

```
/* Function Name: getComputerChoice
*  Description: Determines computer's choice between R, P, or S using a random
number
*  Return Values: 0, 1, or 2
*/
int getComputerChoice() {
    return rand() % 3;
    // Modulo returns random number of 0, 1, or 2
}

/* Function Name: determineWinner
*  Description: Determines the winner of each round based on user and computer
choices
*  Return Values: 0 (Tie), 1 (User wins), -1 (Computer wins)
*/
int determineWinner(int userChoice, int computerChoice) {
    if (userChoice == computerChoice) {
        return 0; // Tie
    } else if ((userChoice == ROCK && computerChoice == SCISSORS) ||
               (userChoice == PAPER && computerChoice == ROCK) ||
               (userChoice == SCISSORS && computerChoice == PAPER)) {
        return 1; // User wins
    } else {
        return -1; // Computer wins
    }
}
```