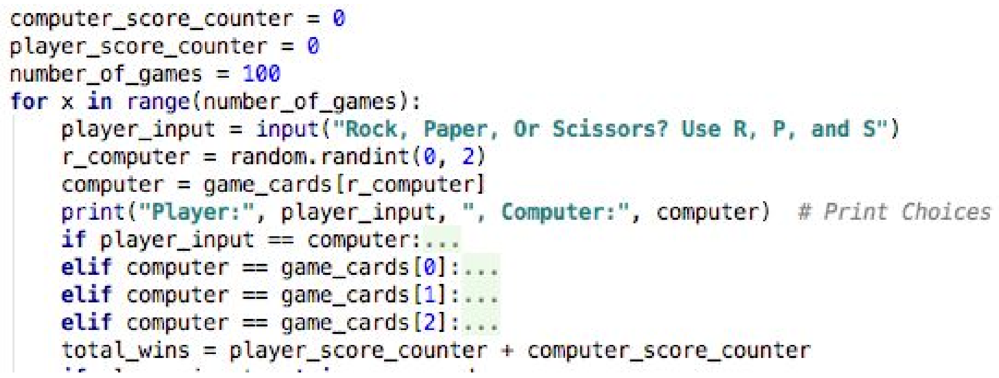
2a.The program is written in python. The purpose of my program is to allow a user and the computer to play the common game of "rock, paper, scissors". The video illustrates how the function prints out the remaining score after the game has finished.

2b. This development was completely independant. I had to analyze how to make the code as simple as possible. I began the code with a introduction to how the game works. That part was simple, but the next part, which was developing the combinations, was tricky. I started off by using if and else statement for each combo, but quickly realized that it was not logical. I decided to use if and else statements based on what the player chose. This strategy used less code and was much simpler. From that point, I had to create a the score counter for both the player and the computer. At first, I used booleans for this section. One of my iterative developments was choosing to change that section to if and else statements because it made the code less complex.

2c.

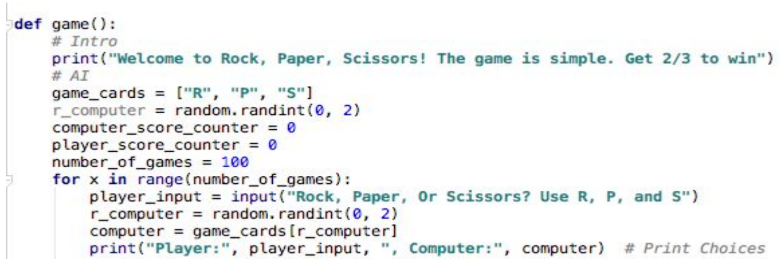


```
computer_score_counter = 0
player_score_counter = 0
number_of_games = 100
for x in range(number_of_games):
    player_input = input("Rock, Paper, Or Scissors? Use R, P, and S")
    r_computer = random.randint(0, 2)
    computer = game_cards[r_computer]
    print("Player:", player_input, ", Computer:", computer)  # Print Choices
    if player_input == computer:...
    elif computer == game_cards[0]:...
    elif computer == game_cards[1]:...
    elif computer == game_cards[2]:...
    total_wins = player_score_counter + computer_score_counter
```

This algorithm counts points for each win using if and else statements.
I used the variable "computer_score_counter" to add up each point from each time the computer won, and "player_score_counter" to add up each

point from each time the player won. I used += to 1 point each time there was a win. I also added the variable "total_wins", and assigned it to add the two previous variables. This combination allows the user to see how much they scored in the game.

2d.



```python
def game():
    # Intro
    print("Welcome to Rock, Paper, Scissors! The game is simple. Get 2/3 to win")
    # AI
    game_cards = ["R", "P", "S"]
    r_computer = random.randint(0, 2)
    computer_score_counter = 0
    player_score_counter = 0
    number_of_games = 100
    for x in range(number_of_games):
        player_input = input("Rock, Paper, Or Scissors? Use R, P, and S")
        r_computer = random.randint(0, 2)
        computer = game_cards[r_computer]
        print("Player:", player_input, ", Computer:", computer)  # Print Choices
```

In my code, I used one abstraction. This abstraction helped to manage the complexity of my code because instead of having to call multiple functions, I called only one, and the game began. My abstraction uses for loops, and this allows the game to repeat multiple times.