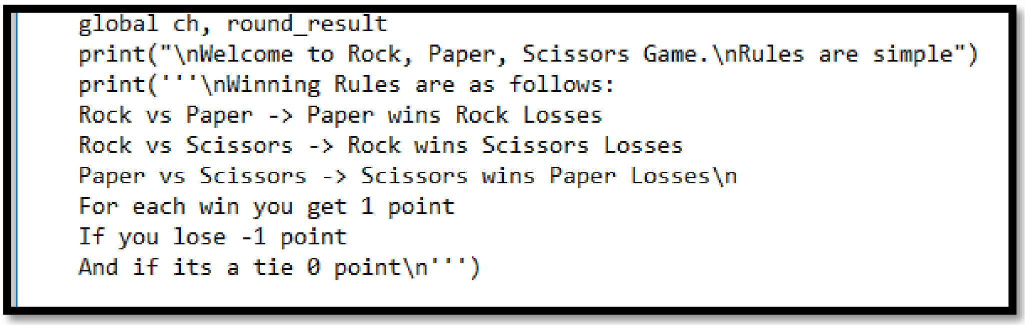
Rock Paper Scissor Game

Write a program that simulates a game of ‘Rock, Paper, Scissors’. You must include the following in your program:

1) Utilize the brainstorm area efficiently - either jotting ideas down, creating an outline, write simple pseudocode (5 pts).

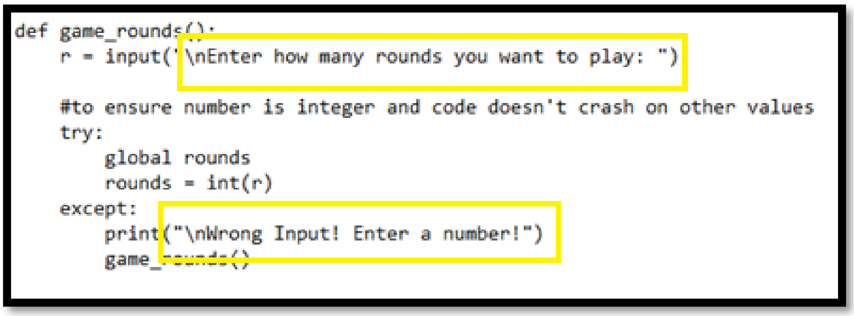
```
Start
Step 1: Initialize all the score and rounds to zero. Create a list to store round results.
Step 2: Take the number of rounds as input, rounds = input()
Step 3: Run the steps 4 to 6 rounds times:
Step 4: Ask the user choice, plr_ch = input() and generate computer choice as comp_ch = random().
Step 5: Check who wins based on the game rules and the entered choice.
Step 6: Update the scores as if win then win+= 1, if loss then loss+=1 else tie+=1.
Step 7: At the end of rounds, display:
    I. total score i.e. wins+loss,
    II. total wins, losses and ties,
    III. and the round wise result list.
End.
```

2) Clear instructions must be given to the user for each step (5 pts)



```
global ch, round_result
print("\nWelcome to Rock, Paper, Scissors Game.\nRules are simple")
print('''\nWinning Rules are as follows:
Rock vs Paper -> Paper wins Rock Losses
Rock vs Scissors -> Rock wins Scissors Losses
Paper vs Scissors -> Scissors wins Paper Losses\n
For each win you get 1 point
If you lose -1 point
And if its a tie 0 point\n''')
```

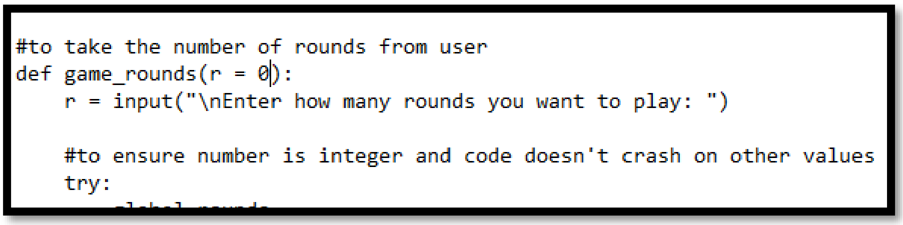
At each step we are telling the user what to input and the input format. Proper instructions are ensured!



```
def game_rounds():
    r = input("\nEnter how many rounds you want to play: ")

    #to ensure number is integer and code doesn't crash on other values
    try:
        global rounds
        rounds = int(r)
    except:
        print("\nWrong Input! Enter a number!")
        game_rounds()
```

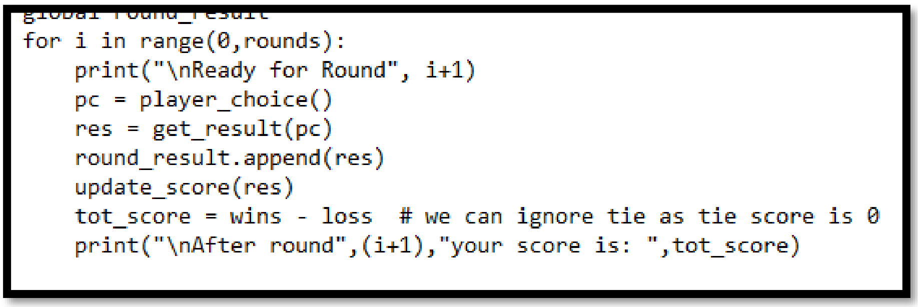
2) At least 1 function with the following requirements (5 pts):

a. At least 1 parameter must be created when defining the function



```
#to take the number of rounds from user
def game_rounds(r = 0):
    r = input("\nEnter how many rounds you want to play: ")

    #to ensure number is integer and code doesn't crash on other values
    try:
        global rounds
```

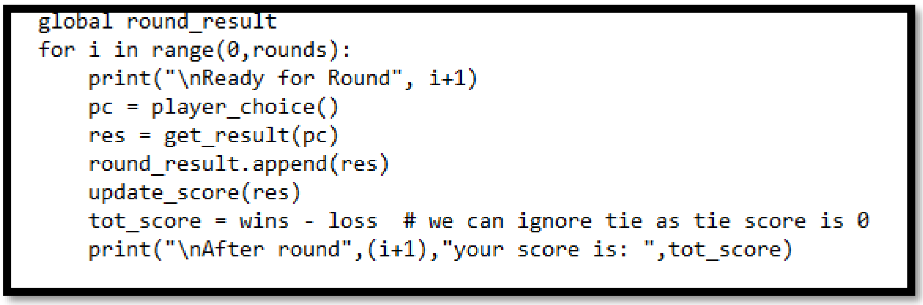
b. Contains sequencing



```
global round_result
for i in range(0,rounds):
    print("\nReady for Round", i+1)
    pc = player_choice()
    res = get_result(pc)
    round_result.append(res)
    update_score(res)
    tot_score = wins - loss  # we can ignore tie as tie score is 0
    print("\nAfter round",(i+1),"your score is: ",tot_score)
```

Sequence is defined by the call of functions so *player_choice*() is executed before *get_result*() and *update_score*() is executed after.

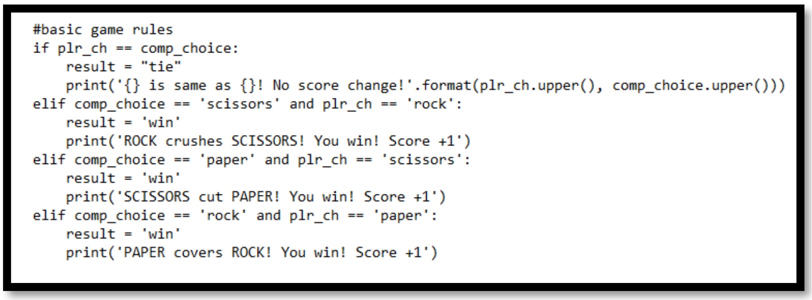
c. Contains iteration



```
global round_result
for i in range(0,rounds):
    print("\nReady for Round", i+1)
    pc = player_choice()
    res = get_result(pc)
    round_result.append(res)
    update_score(res)
    tot_score = wins - loss  # we can ignore tie as tie score is 0
    print("\nAfter round",(i+1),"your score is: ",tot_score)
```

For loop is a good example of iteration.

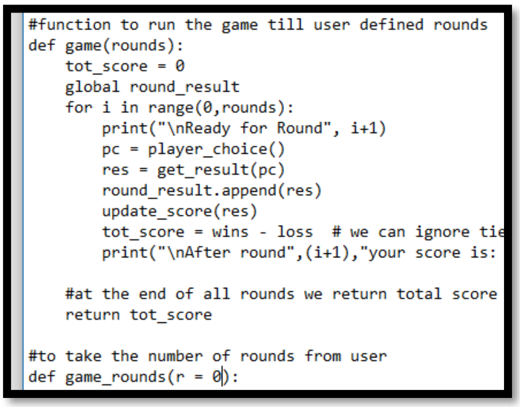
d. Contains selection



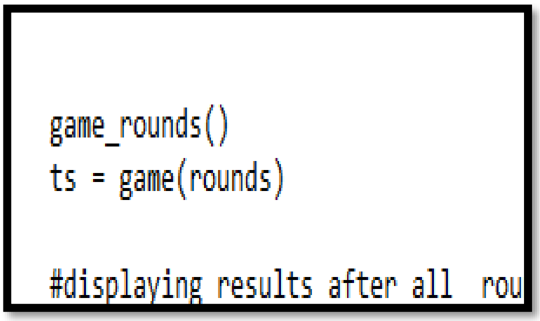
```
#basic game rules
if plr_ch == comp_choice:
    result = "tie"
    print('{} is same as {}! No score change!'.format(plr_ch.upper(), comp_choice.upper()))
elif comp_choice == 'scissors' and plr_ch == 'rock':
    result = 'win'
    print('ROCK crushes SCISSORS! You win! Score +1')
elif comp_choice == 'paper' and plr_ch == 'scissors':
    result = 'win'
    print('SCISSORS cut PAPER! You win! Score +1')
elif comp_choice == 'rock' and plr_ch == 'paper':
    result = 'win'
    print('PAPER covers ROCK! You win! Score +1')
```

Here the further execution process is determined by the selection statements if else.

3) Call the function that you created in #2 (5 pts)



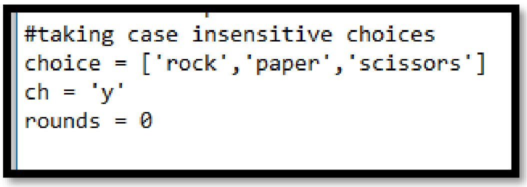
```
#function to run the game till user defined rounds
def game(rounds):
    tot_score = 0
    global round_result
    for i in range(0,rounds):
        print("\nReady for Round", i+1)
        pc = player_choice()
        res = get_result(pc)
        round_result.append(res)
        update_score(res)
        tot_score = wins - loss  # we can ignore tie
        print("\nAfter round",(i+1),"your score is:

    #at the end of all rounds we return total score
    return tot_score

#to take the number of rounds from user
def game_rounds(r = 0):
```



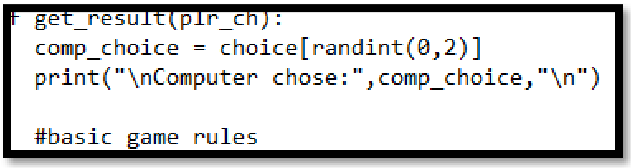
```
game_rounds()
ts = game(rounds)

#displaying results after all  rou
```

We can change the call based on our requirement or the default value we want for the rounds. Here we want to default to zero so no *game_rounds*() is called without a number, else we can specify.

4) Use the list that was provided to you in a way that manages the complexity of your program (utilizing the list should have made it easier for you to program your game). (5 pts)



```
#taking case insensitive choices
choice = ['rock','paper','scissors']
ch = 'y'
rounds = 0
```

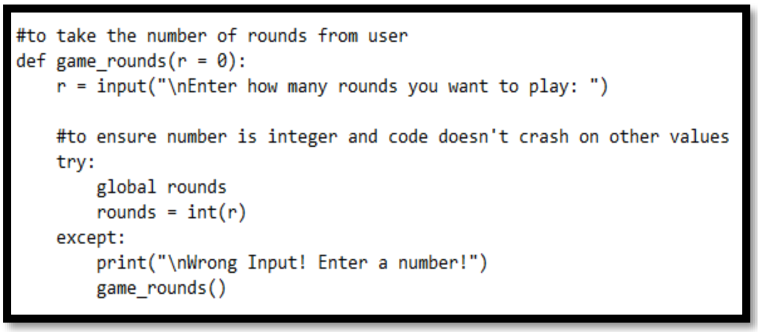


```
f get_result(plr_ch):
  comp_choice = choice[randint(0,2)]
  print("\nComputer chose:",comp_choice,"\n")

  #basic game rules
```

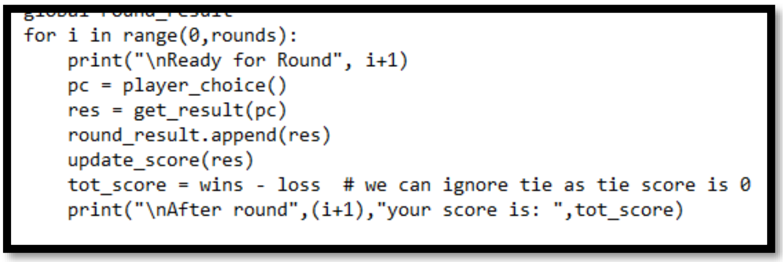
Using the list reduced the complexity else we would have to map a random number to the options rock, paper and scissor and then use if-else ladder increasing space and time complexities.

5) The user should be able to put how many rounds they want to play against the ‘other player’ in the beginning and the program should repeat that many times (5 pts).

Here the user can input the number of rounds that has to be a number and the for loop will ensure code runs that many times.



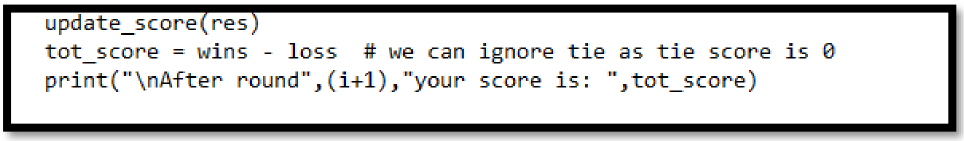
```
#to take the number of rounds from user
def game_rounds(r = 0):
    r = input("\nEnter how many rounds you want to play: ")

    #to ensure number is integer and code doesn't crash on other values
    try:
        global rounds
        rounds = int(r)
    except:
        print("\nWrong Input! Enter a number!")
        game_rounds()
```



```
global round_result
for i in range(0,rounds):
    print("\nReady for Round", i+1)
    pc = player_choice()
    res = get_result(pc)
    round_result.append(res)
    update_score(res)
    tot_score = wins - loss  # we can ignore tie as tie score is 0
    print("\nAfter round",(i+1),"your score is: ",tot_score)
```

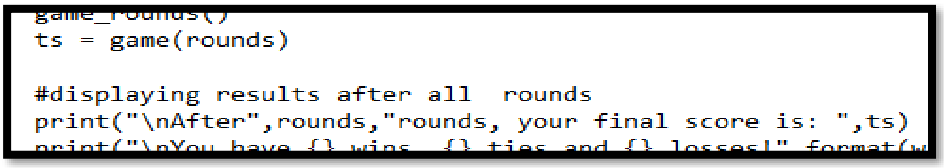
6) Your program should either have a running total after each round and/or show the final score at the end (user vs ‘other player’) (5 pts)

Running total: *tot_score* is the score after each round.



```
update_score(res)
tot_score = wins - loss  # we can ignore tie as tie score is 0
print("\nAfter round",(i+1),"your score is: ",tot_score)
```

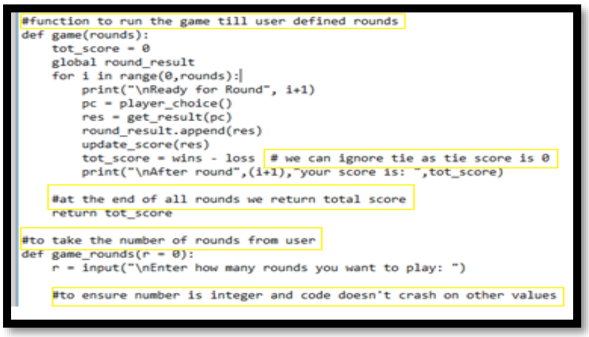
Final Score: ts is the final score after all the rounds.



```
game_rounds()
ts = game(rounds)

#displaying results after all  rounds
print("\nAfter",rounds,"rounds, your final score is: ",ts)
print("\nYou have {} wins, {} ties and {} losses!".format(w
```

7) You must include at least 5 comments where you are explaining parts of your code and reasoning and commenting on the parts that you and your partner made. (5 pts)

The below screenshot of a part of code alone has 5 comments the code has many more.



```
#function to run the game till user defined rounds
def game(rounds):
    tot_score = 0
    global round_result
    for i in range(0,rounds):
        print("\nReady for Round", i+1)
        pc = player_choice()
        res = get_result(pc)
        round_result.append(res)
        update_score(res)
        tot_score = wins - loss  # we can ignore tie as tie score is 0
        print("\nAfter round",(i+1),"your score is: ",tot_score)

    #at the end of all rounds we return total score
    return tot_score

#to take the number of rounds from user
def game_rounds(r = 0):
    r = input("\nEnter how many rounds you want to play: ")

    #to ensure number is integer and code doesn't crash on other values
```

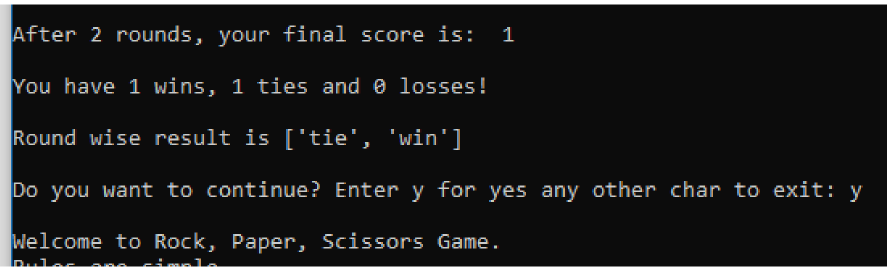
8) Your program cannot be broken/crashed at any point; it should be id10t-proof (15 pts)

At all the input points and execution sequences we have checked for the mishap cases, and used either try exception to handle erroneous cases or used if else to validate the input to ensure no crash.

Extra Credit Options

- The program should ask if the user would like play again before terminating by using another function (1 pt)

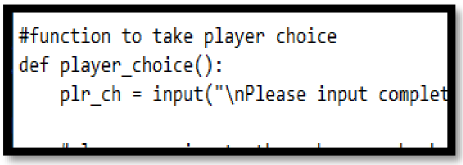
After all the rounds: If user inputs y, it will continue else exit.



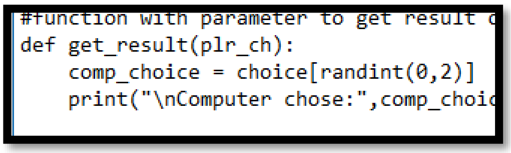
```
After 2 rounds, your final score is:  1

You have 1 wins, 1 ties and 0 losses!

Round wise result is ['tie', 'win']

Do you want to continue? Enter y for yes any other char to exit: y

Welcome to Rock, Paper, Scissors Game.
Rules are simple
```

- Use at least 3 functions (including the required one so 2 additional ones – these don't necessarily need parameters) (1 pt)

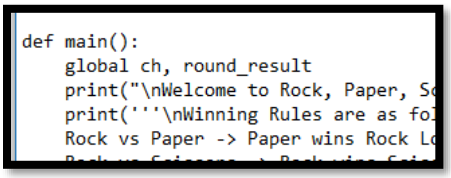
There are many functions, some are shown below:



```
#function to take player choice
def player_choice():
    plr_ch = input("\nPlease input complet
```



```
#function with parameter to get result c
def get_result(plr_ch):
    comp_choice = choice[randint(0,2)]
    print("\nComputer chose:",comp_choic
```



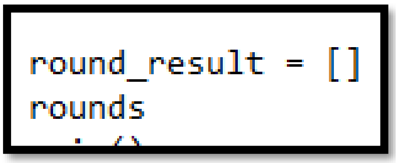
```
def main():
    global ch, round_result
    print("\nWelcome to Rock, Paper, Sc
    print('''\nWinning Rules are as fol
    Rock vs Paper -> Paper wins Rock Lo
    Rock vs Scissors -> Rock wins Scis
```

- Have zero While True loops inside of While True loops (1 pt)

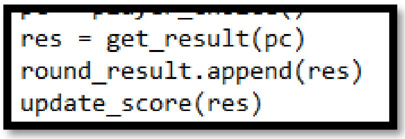
The whole code does not have any while(True) loops.

- Create and use an additional significant list (this purpose of the additional list is crucial to your game) (2 pts)

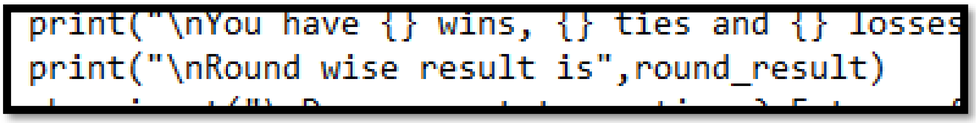
The list *round_result*[] stores result of each rounds to display at the end.



```
round_result = []
rounds
```



```
res = get_result(pc)
round_result.append(res)
update_score(res)
```



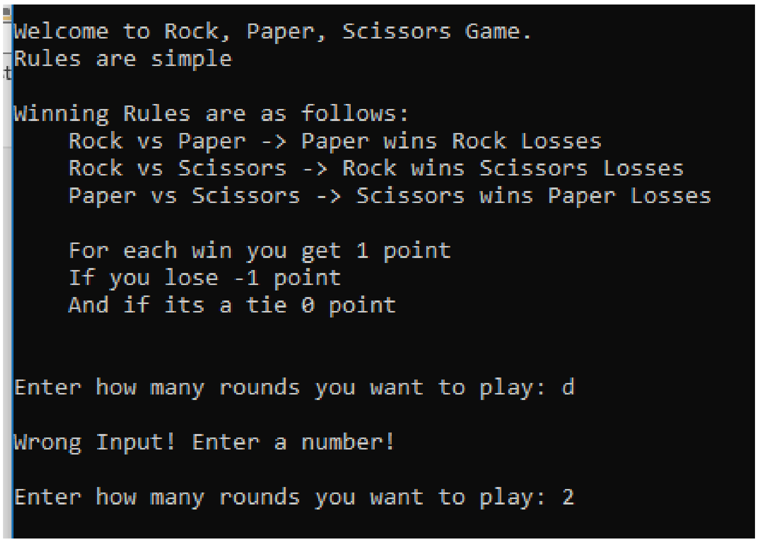
```
print("\nYou have {} wins, {} ties and {} losses
print("\nRound wise result is",round_result)
```

Source Code:

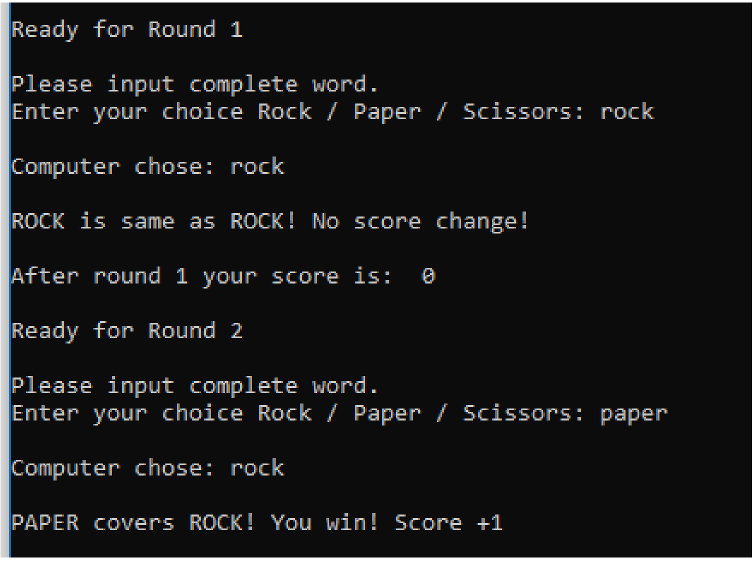
```
from random import randint
#taking case insensitive choices
choice = ['rock','paper','scissors']
ch = 'y'
rounds = 0

#function to take player choice
def player_choice():
    plr_ch = input("\nPlease input complete word.\nEnter your choice Rock / Paper / Scissors: ")

    #player can input other char so check to ensure that cannot be broken
    if plr_ch.lower() and plr_ch.lower() in ('rock','paper','scissors'):
        return plr_ch.lower()
    else:
        print("\nWrong choice!! Retry !!")
        player_choice()

#function with parameter to get result on every choice
def get_result(plr_ch):
    comp_choice = choice[randint(0,2)]
    print("\nComputer chose:",comp_choice,"\n")
    #basic game rules
    if plr_ch == comp_choice:
        result = "tie"
        print('{} is same as {}! No score change!'.format(plr_ch.upper(), comp_choice.upper()))
    elif comp_choice == 'scissors' and plr_ch == 'rock':
        result = 'win'
        print('ROCK crushes SCISSORS! You win! Score +1')
    elif comp_choice == 'paper' and plr_ch == 'scissors':
        result = 'win'
        print('SCISSORS cut PAPER! You win! Score +1')
    elif comp_choice == 'rock' and plr_ch == 'paper':
        result = 'win'
        print('PAPER covers ROCK! You win! Score +1')

    #if it does not match any of the win criteria then add 1 to loss and
    #display loss message
    else:
        result = 'lose'
        print('You lose! Score -1')
    return result

#function to update the scores
def update_score(result):
    global wins, loss, tie
    if result == 'win':
        wins += 1
    elif result == 'lose':
        loss += 1
    else:
        tie += 1
```

```
#function to run the game till user defined rounds
def game(rounds):
    tot_score = 0
    global round_result
    for i in range(0,rounds):
        print("\nReady for Round", i+1)
        pc = player_choice()
        res = get_result(pc)
        round_result.append(res)
        update_score(res)
        tot_score = wins - loss  # we can ignore tie as tie score is 0
        print("\nAfter round",(i+1),"your score is: ",tot_score)

    #at the end of all rounds we return total score
    return tot_score

#to take the number of rounds from user
def game_rounds(r = 0):
    r = input("\nEnter how many rounds you want to play: ")

    #to ensure number is integer and code doesn't crash on other values
    try:
        global rounds
        rounds = int(r)
    except:
        print("\nWrong Input! Enter a number!")
        game_rounds()

def main():
    global ch, round_result
    print("\nWelcome to Rock, Paper, Scissors Game.\nRules are simple")
    print('''\nWinning Rules are as follows:
    Rock vs Paper -> Paper wins Rock Losses
    Rock vs Scissors -> Rock wins Scissors Losses
    Paper vs Scissors -> Scissors wins Paper Losses\n
    For each win you get 1 point
    If you lose -1 point
    And if its a tie 0 point\n''')



    game_rounds()
    ts = game(rounds)

    #displaying results after all  rounds
    print("\nAfter",rounds,"rounds, your final score is: ",ts)
    print("\nYou have {} wins, {} ties and {} losses!".format(wins,tie,loss))
    print("\nRound wise result is",round_result)
    ch = input("\nDo you want to continue? Enter y for yes any other char to exit: ")

while(ch == 'y' or ch == 'Y'):
    wins = 0
    loss = 0
    tie = 0

    round_result = []
    rounds
    main()
print("\nSee you Again!!")
```

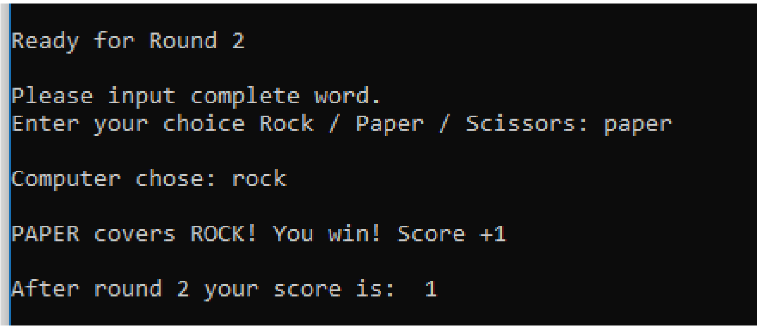
Output:



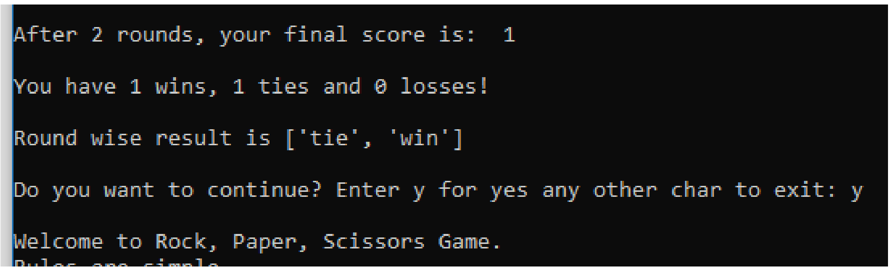
```
Welcome to Rock, Paper, Scissors Game.
Rules are simple

Winning Rules are as follows:
    Rock vs Paper -> Paper wins Rock Losses
    Rock vs Scissors -> Rock wins Scissors Losses
    Paper vs Scissors -> Scissors wins Paper Losses

    For each win you get 1 point
    If you lose -1 point
    And if its a tie 0 point


Enter how many rounds you want to play: d

Wrong Input! Enter a number!

Enter how many rounds you want to play: 2
```



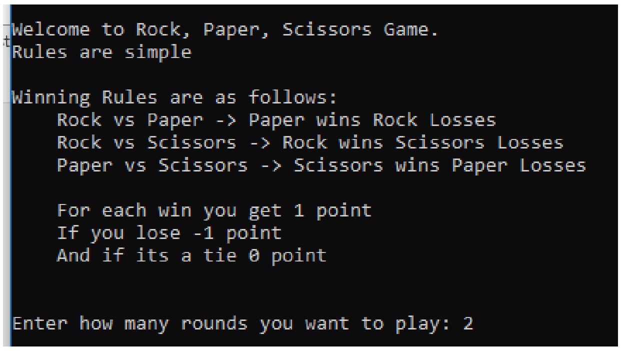
```
Ready for Round 1

Please input complete word.
Enter your choice Rock / Paper / Scissors: rock

Computer chose: rock

ROCK is same as ROCK! No score change!

After round 1 your score is:  0

Ready for Round 2

Please input complete word.
Enter your choice Rock / Paper / Scissors: paper

Computer chose: rock

PAPER covers ROCK! You win! Score +1
```



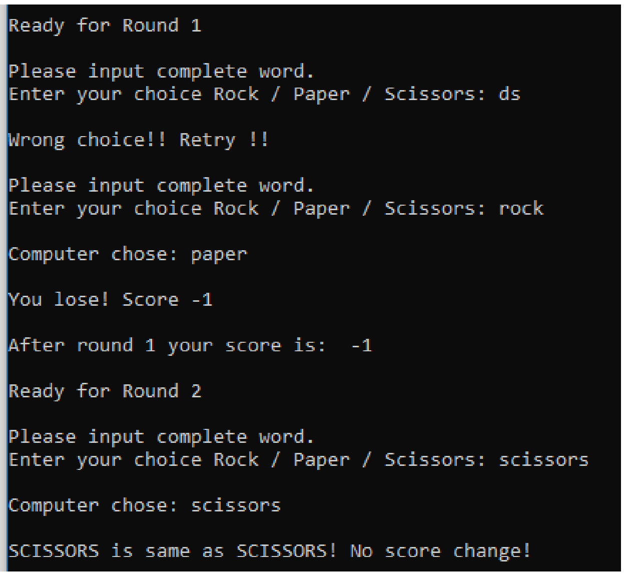
```
Ready for Round 2

Please input complete word.
Enter your choice Rock / Paper / Scissors: paper

Computer chose: rock

PAPER covers ROCK! You win! Score +1

After round 2 your score is:  1
```



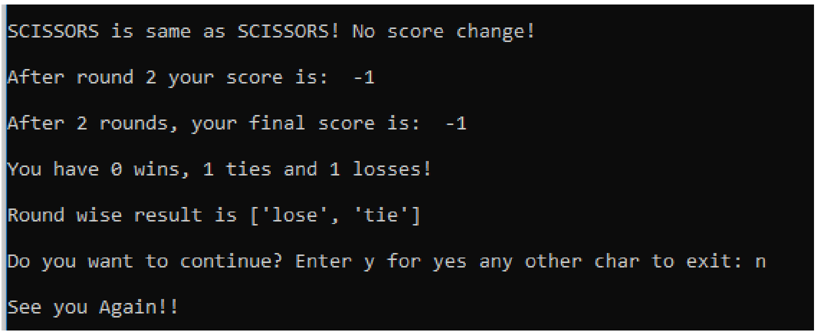
```
After 2 rounds, your final score is:  1

You have 1 wins, 1 ties and 0 losses!

Round wise result is ['tie', 'win']

Do you want to continue? Enter y for yes any other char to exit: y

Welcome to Rock, Paper, Scissors Game.
Rules are simple
```



```
Welcome to Rock, Paper, Scissors Game.
Rules are simple

Winning Rules are as follows:
    Rock vs Paper -> Paper wins Rock Losses
    Rock vs Scissors -> Rock wins Scissors Losses
    Paper vs Scissors -> Scissors wins Paper Losses

    For each win you get 1 point
    If you lose -1 point
    And if its a tie 0 point


Enter how many rounds you want to play: 2
```



```
Ready for Round 1

Please input complete word.
Enter your choice Rock / Paper / Scissors: ds

Wrong choice!! Retry !!

Please input complete word.
Enter your choice Rock / Paper / Scissors: rock

Computer chose: paper

You lose! Score -1

After round 1 your score is:  -1

Ready for Round 2

Please input complete word.
Enter your choice Rock / Paper / Scissors: scissors

Computer chose: scissors

SCISSORS is same as SCISSORS! No score change!
```



```
SCISSORS is same as SCISSORS! No score change!

After round 2 your score is:  -1

After 2 rounds, your final score is:  -1

You have 0 wins, 1 ties and 1 losses!

Round wise result is ['lose', 'tie']

Do you want to continue? Enter y for yes any other char to exit: n

See you Again!!
```