1. Develop the classic game rock-paper-scissors in Python (100pts)

(a) Introduction

It is a simultaneous game with two players. Each player chooses one of rock, paper or scissors, without knowing the other one's choice. The rules are: rock beats scissors; scissors beat paper; and paper beats rock. It is a tie if both players make the same choice.

(b) The Task

Several functions have been defined with clear descriptions of parameters and returns, which will help you implement the game. Please read the file "Assignment1_Rock_ Paper_Scissors.py" for the details. Your task is to implement those functions in that file.

HumanPlayer	20pts
ComputerPlayer	10pts
Judge	10pts
PrintOutcome	10pts
UpdateGameRecord	10pts
PrintGameRecord	10pts
PlayGame	30pts

There are two players: one player is a human user represented by the function "HumanPlayer", and the other player is the function "ComputerPlayer".

The function HumanPlayer will parse user input from the computer keyboard, and it should be able to handle random input string from the human user. The human user may also choose to print the record of the game or quit.

The function "ComputerPlayer" will use a random strategy: select rock/paper/scissors randomly. You will need to import the package random to use the function randint. See the document: https://docs.python.org/3/library/random.html

The outcome of a game will be determined by the function "Judge" and printed to the console. The game will be played many rounds until the human user wants to quit.

A list "RecordOfGame" will be used to keep a record of players' choices and outcomes. It will be updated after two players have made the choices at each round.

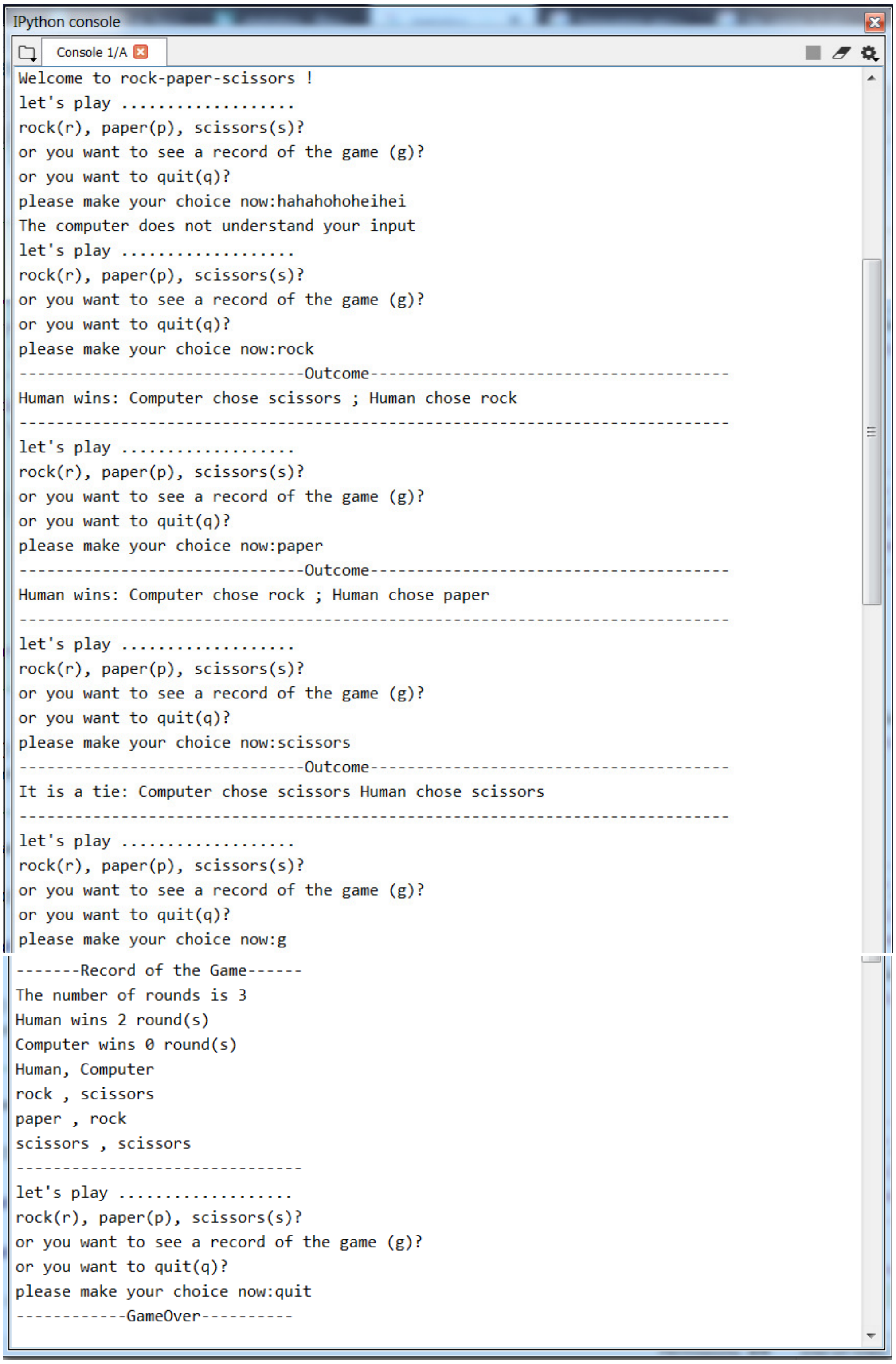
(c) Notes

Do not use any global variables (variables outside the functions) because they are evil.
Do not change the definitions of the functions in "Project_Rock_Paper_Scissor.py"
Do not define any new functions because the provided functions are enough for the game.

Read the sample run in the next page carefully. You are free to use any words in the messages as long as they provide the same information to the user.

Play your game many times to make sure it has no bugs.

(d) Sample Run



```
Welcome to rock-paper-scissors !
let's play ..................
rock(r), paper(p), scissors(s)?
or you want to see a record of the game (g)?
or you want to quit(q)?
please make your choice now:hahahohoheihei
The computer does not understand your input
let's play ..................
rock(r), paper(p), scissors(s)?
or you want to see a record of the game (g)?
or you want to quit(q)?
please make your choice now:rock
-------------------------------Outcome--------------------------------------
Human wins: Computer chose scissors ; Human chose rock
----------------------------------------------------------------------------
let's play ..................
rock(r), paper(p), scissors(s)?
or you want to see a record of the game (g)?
or you want to quit(q)?
please make your choice now:paper
-------------------------------Outcome--------------------------------------
Human wins: Computer chose rock ; Human chose paper
----------------------------------------------------------------------------
let's play ..................
rock(r), paper(p), scissors(s)?
or you want to see a record of the game (g)?
or you want to quit(q)?
please make your choice now:scissors
-------------------------------Outcome--------------------------------------
It is a tie: Computer chose scissors Human chose scissors
----------------------------------------------------------------------------
let's play ..................
rock(r), paper(p), scissors(s)?
or you want to see a record of the game (g)?
or you want to quit(q)?
please make your choice now:g
-------Record of the Game------
The number of rounds is 3
Human wins 2 round(s)
Computer wins 0 round(s)
Human, Computer
rock , scissors
paper , rock
scissors , scissors
-------------------------------
let's play ..................
rock(r), paper(p), scissors(s)?
or you want to see a record of the game (g)?
or you want to quit(q)?
please make your choice now:quit
------------GameOver----------
```

Have fun in programming!

2. Enhance the computer player for the game rock-paper-scissors (Extra Credit 20pts)

The goal is to enhance "ComputerPlayer" so that it could beat the human user with a high probability. There are many strategies to achieve this goal, and the simplest one is called weighted random choice, which you will implement for this task.

Here is the detailed description of the algorithm. The game may have been played many rounds between ComputerPlayer and HumanPlayer. ComputerPlayer will calculate the frequencies of the human user's choices using GameRecord.

Let R be the number of "rock" that the human user has chosen

Let P be the number of "paper" that the human user has chosen

Let S be the number of "scissors" that the human user has chosen

Then R+P+S is the number of rounds.

The frequency of "rock" is R/(R+P+S).

The frequency of "paper" is P/(R+P+S).

The frequency of "scissors" is S/(R+P+S).

Based on the above information, the ComputerPlayer will make a weighted random choice. If the frequencies of the human user's choices are [0.1(rock), 0.2(paper), 0.7(scissors)], then ComputerPlayer should use the frequencies of [0.7(rock), 0.1(paper), 0.2(scissors)].

You need to import numpy package, and use the choice function:

```
import numpy as np

draw = np.random.choice(['rock', 'paper', 'scissors'], 1, p=[0.7, 0.1, 0.2])[0]
```

Note:

ComputerPlayer should not "look" at the current choice of the human player.

You are free to add new functions (e.g. to calculate the frequencies)

Save your code to a different file: Project_Rock_Paper_Scissor_enhanced_1.py

3. Further enhancement of the computer player (Extra Credit 20pts)

A human user may have some favorite combinations of actions, such as "rock" followed by "paper". Instead of counting frequencies of individual choices/actions, the ComputerPlayer may use frequencies of action pairs to compete against the user.

Your task is to implement this strategy.

Note: add comments in your program to clearly explain your algorithms.

Save your code to a different file: Project_Rock_Paper_Scissor_enhanced_2.py