

## Read me

### U24567277 Shifeng LI

In my code, I use the random probability to choose scissor, rock and paper. Firstly, open the original data, the name of the file is "DATA", and transform the data to be the integer number. I use 1 to express Scissor, use 2 to express rock and use 3 to express paper. After running program, I can get the result like that:

```
[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
[2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0]
[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
[2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0]
[3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0]
[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
32
34
34
```

```
computer.append([1,1])
return person1, person2, person3, computer1, computer2, computer3, person_win, computer_win, both_tie
if __name__ == '__main__':
```

When I running the program at first time, I use random number to be initialization. After that, compute the number of person winning and losing in original data, and I also choose the random number which is based on the winning and losing number to be the computer' s choice. I separate the winning number into three parts, such as the number of using rock.

I use a math function to compute the number and get the result , like that:

```
print('computer is' + str(select[computer -
result = (user - computer + 4) % 3 - 1
```

If the result >0 then person win. If the result =0 then tie. If the result <0 then computer win.

```
/Users/airwolf/PyCharm
2
you areR
computer isP
Computer Win
Continue? [Y/N]:
```

I use this method and test it get the result, and find it is better than original data, and use a file named "My DATA" to record them,

```
[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
[2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0]
[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
[2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0]
[3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0]
[1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
50
25
25
```