Homework 2
CmpE 150, Introduction to Computing, Spring 2023

Instructor: Mehmet Turan
TAs: Kutay Altıntaş, Mehmet Özer
SA: Emre Kılıç

Due: 09/04/2023, 23:59

1 Introduction

Long ago, in a kingdom far away, there lived a wise old sage. He was renowned throughout the land for his wisdom and his knowledge of strategy. One day, the cruel king of the land summoned the old sage to his court and asked him to invent a game that would satisfy himself. He wanted a game that would challenge his subjects and allow them to showcase their intelligence and skill against his superior mind.

Sage wanted to teach him a lesson so he took up the challenge and spent many years perfecting his game. He called it Blackking, a game that required a combination of strategy and luck to win. The game was played with a deck of cards, and the goal was to reach a hand value of 21 or as close to 21 as possible, without going over.

The king loved the game and asked the sage if he wished anything. That was the time to teach him a lesson and sage challenged the king and put some pot. Every time he wins, king had to pay 50 golds but every time the sage loses, he had to pay 50 golds back.

2 Details

In this homework you are asked to implement Blackking where user is the sage and plays against the king.

In the given .py file there will be a function get_next_card() that returns random cards from the deck. The card format is a 2-tuple where the first element is the suit of the card and second one is the point of the card. There will be 4 suits (Spades, Hearts, Clubs, Diamonds). 2 of them (Spades and Clubs) are black-colored and other two are red-colored. You will use this function to draw a card.

Firstly, you will take an integer input with this prompt "Sage's money: " and this input will be the beginning pot. Secondly, you will take another integer input with this prompt "Number of games: " and this input will be the number of games that will be played.

In each game, at the start, you will print out the game number i.e print "Game 1:" etc. You will draw 2 cards for the sage and 2 cards for the king, respectively. Firstly, print "King's cards: " and print out the king's cards but you have to hide the first card i.e print "(*)" for the the first card and fully print the second card, then print "Total value: " and print out the value of the king's card without the secret one. After that, print "Sage's cards: " and print out all two of the sage's cards. Finally in the fourth line print "Total value: " and print out the total value of the sage. All values for the cards are given in a table below. It is recommended to define a function that returns the value according to its suit and point.

Card	**Value**
A	10
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
J	10
Q	10
K	1 if card is red, 11 if card is black

Table 1: Cards and Their Values

Then, consider total value of the sage, there will be some cases.

1. Total value == 21: That means sage has won the game and earns 50 golds. Print out "It is Blackking! You won!".

2. Total value > 21: That means sage has lost the game and loses 50 golds. Print out "You busted! You lost!".

3. Total value < 21: Take input from the user for further decisions with this prompt "Do you want to hit or stand? [H/S] ". There are three cases according to the decision of the sage.

   Case 1. The input is "H": Draw a card for the sage and print out the state of the sage.

* If the new total value == 21 then stop.
* If the new total value > 21 then sage has lost the game and loses 50 golds. Print out "You busted! You lost!".
* If the new total value < 21 then repeat the process. Take an input with this prompt "Do you want to hit or stand? [H/S] ".

Case 2. The input is "S": Then stop.

After you stopped, now its king's turn. Print out king's state without hiding the first card. Show its full deck and total value. Note that if the game ends in the sage's turn, you must not consider king's deck. Now there will be some cases too.

1. Total value > 21: That means sage has won the game and earns 50 golds. Print out "King busted! You won!".
2. Total value ≥ 17: That means king has stopped and now look at the values of both sides.
   * If King has higher value then sage has lost the game and loses 50 golds. Print out "King has higher value. You lost!".
   * If Sage has higher value then sage has won the game and earns 50 golds. Print out "You have higher value. You won!".
   * If there is a tie then no one wins the game and money remains the same. Print out "It is a tie!".
3. Total value < 17: That means king hits and draw one card from the deck. Print out the king's state and repeat this process again.

After all the games are played, print "Final money is " and print out the remaining money of the sage.

3 Input & Output

3.1 Input

First line is the pot that the sage put in. Second line is the number of games the sage plays against the king. Remaining lines are the moves of the sage.

Input	**Explanation**
100	Amount of gold that sage puts in
3	Number of games the sage plays against the king
H	First decision
H	Second decision
S	Third decision

Table 2: Sample Input with Explanations

3.2 Output

Last output is the final money that the sage have. Former outputs are the flow of the game.

Output	**Explanation**
Sage's money:	Prompt for user to enter sage's initial gold
Number of games:	Prompt for user to enter number of games
Game 1	Game 1 Flow
King's cards: (*)('Diamonds', '3')	
Total value: 3	
Sage's cards: ('Clubs', '3')('Spades', 'K')	
Total value: 14	
Do you want to hit or stand? [H/S]	Prompt for user to select sage's action
Sage's cards: ('Clubs', '3')('Spades', 'K')('Clubs', 'K')	
Total value: 24	
You busted! You lost!	Sage's total value exceeds 21
Game 2	Game 2 Flow
King's cards: (*)('Spades', '2')	
Total value: 2	
Sage's cards: ('Spades', '8')('Spades', '6')	
Total value: 14	
Do you want to hit or stand? [H/S]	Sage selects 'H'
Sage's cards: ('Clubs', '3')('Spades', 'K')('Diamonds', 'J')	
Total value: 24	
You busted! You lost!	Sage's total value exceeds 21
Game 3	Game 3 Flow
King's cards: (*)('Clubs', '10')	
Total value: 10	
Sage's cards: ('Clubs', '8')('Spades', 'A')	
Total value: 18	
Do you want to hit or stand? [H/S]	Sage selects 'S', thus it is King's turn to play
King's cards: ('Spades', '5')('Clubs', '10')	King's second card is revealed.
Total value: 15	Total value < 17, King hits.
King's cards: ('Spades', '5')('Clubs', '10')('Hearts', '8')	King's third card is ('Hearts', '8').
Total value: 23	
King busted! You won!	King has a value > 21, sage wins.
Final money is 50	Final money of sage

Table 3: Sample Output with Explanations

4 Submission

Your code will be graded automatically. Therefore, it's important that you follow the submission instructions. **You will lose points if your submission does not comply with the submission rules.**

First, your solution Python script should be put in a folder named **homework2**. Then, you should zip the **homework2** folder and rename it to <**student**$_{id}$>**.zip**. This zip file will be submitted through Moodle. Name of your solution script must be **solution.py**.

5 Grading

The grading of this assignment is based on the automatic run and the success of your code in test cases. Final grade will be the sum of collected points from each test case.

6 Warnings

1. This is an individual homework.
2. All scripts are checked automatically for similarity with other submissions and exercises from previous years. Do not copy codes from the internet or your friends. Make sure you write and submit your own code. Any sign of cheating will be penalized, and you will get **-50** points for the homework, and you will get F grade in case of recurrence in any other homework.