Assignment 2
Deadline - 29th August 2021, 11.59 PM

Instructions:

a. Create a folder named Assignment2_Group_Number. Inside it, keep the python code solving the problem, the templates folder (if using flask) containing the required html file and the static folder (if using flask) containing the utility.js file (same as what you use everyday). Submit from one of the group member's accounts.

b. Create a .zip of the entire folder and submit in piazza in "Assignment 2" folder.

c. Flask is encouraged but not mandatory for this assignment.

Problem

Design a four player card game using Python in the following manner.

Setup:

There are a total of 52 cards distributed into four suits - spades, hearts, diamonds and clubs. Each set has 13 cards - [Ace, King, Queen, Joker, 10, 9, 8, 7, 6, 5, 4, 3, 2] arranged in the descending order of weightage i.e. "Ace" is the most powerful card whereas "2" is the least powerful one.

It would be a four player game where the user will be playing with three other computer simulated bots.

Steps of the game:

1. FIrst, distribute all the cards randomly to all four of the players.

2. The user can see only his/her cards.

3. Each player makes a "call" of how many turns they can potentially win. The bots place the "call" in the following simple manner - they check how many of "Ace","King","Queen" and "Joker" cards they have from each suit and give those many "calls". For instance, if a bot has the "Ace" and "King" of spades and "Queen" of hearts, then it would give "call" = 2+1=3. User can give calls as per his/her choice.

4. After "calls" are made, the game starts from a random player and goes in a clockwise fashion. You can choose where you want to place the user in the circle.

5. If a bot is the first player to play in a turn - it will choose the highest weighted card it has and play it. If the user is the first player, he/she can choose any of his/her cards. Before

playing the card, the user will be shown the cards he/she possesses at that point, from each suit.

6. For the next player onwards, if a bot is playing, it will choose a higher weighted card if available to it from the same suit, otherwise it will give the lowest weighted card from the same suit. If cards from that particular suit are not available, it will choose the lowest weighted card available to it, from any other suit. Example - if the player1 has played “King” of “Spades”, if the player2 has “Ace” of “Spades”, it will play it; otherwise it will play the lowest weighted card it has from “Spades”; if no “Spades” card is available to it, the lowest weighted card will be played.
   If the user is playing, he/she can choose the card himself/herself.

7. After each turn is over, decide the winner as the player playing the highest weighted card of the suit the turn had begun with. The winner starts the next turn.

8. Continue the game until all cards are played.

9. Score the players as the following - if the player has given a “call” for winning x number of turns and actually won y number of turns - then, if x> y, score=-10x, otherwise, score=10x+(y-x)

10. Ask the user if he/she would like to continue. If the user wants to finish it, show the final scores.

11. **FILE I/O Requirement -** Apart from generating random card distribution and seating arrangement, there should be an option to read them from one input file.

Sample I/O:

The following lines are printed while the program runs, user input is marked with (IN)

LET FOLLOWING BE THE RANDOM DISTRIBUTION (User should be able to see only his/her cards)

```
bot1:   HA,SK,DQ,CJ,H10,S9,D8,C7,H6,H5,H4,H3,H2
bot2:   HK,SA,DJ,CQ,S10,D9,S8,S7,S6,S5,S4,S3,S2
bot3:   HQ,SJ,DA,CK,C10,C9,C8,H7,C6,C5,C4,C3,C2
player:HJ,SQ,DK,CA,D10,H9,H8,D7,D6,D5,D4,D3,D2
```

Where H - Hearts, C - Clubs, D - Diamonds and S - Spades
Similarly, A - Ace, K - King, Q - Queen, J - Joker

calls of players bot1->4, bot2->4, bot3->4, player->4 (IN)

cyclic order bot1->bot2->player->bot3->bot1………

```
start form bot2
Turn1:              Turn 2:             Turn  3:            Turn 4:             Turn 5:
bot2->SA            bot2->HK            bot1->SK            bot1->DQ            bot3->CK
player->D2 (IN)     player->D3 (IN)     bot2->S2            bot2->D9            bot1->C7
bot3->SJ            bot3->H7            player->D4 (IN)     player->D5 (IN)     bot2->CQ
bot1->S9            bot1->HA            bot3->C2            bot3->DA            player->D6(IN)

bot2 wins           bot1 wins           bot1 wins           bot3 wins           bot3 wins

Turn 6:             Turn 7:             Turn 8:             Turn 9:             Turn 10:
bot3->HQ            bot3->C10           bot1->H10           player->SQ (IN)     player->DK(IN)
bot1->H2            bot1->CJ            bot2->S5            bot3->C4            bot3->C5
bot2->S3            bot2->S4            player->HJ (IN)     bot1->H3            bot1->D8
player->D7 (IN)     player->H8 (IN)     bot3->C3            bot2->S6            bot2->DJ

bot3 wins           bot1 wins           player wins         player wins         player wins


Turn 11:            Turn 12:            Turn 13:
player->CA (IN)     player->D10 (IN)    player->H9 (IN)
bot3->C6            bot3->C8            bot3->C9
bot1->H4            bot1->H5            bot1->H6
bot2->S7            bot2->S8            bot2->S10

player wins         player wins         player wins


scores:
bot1 = -10*4 ( y = 3, x=4)
bot2 = -10*4 ( y = 1, x=4)
bot3 = -10*4 ( y = 3, x=4)
player = 10*4+(6-4) = 42  (y = 6, x = 4)
player is the winner!!!!!!!

Continue(Y/N): Y
```

LET FOLLOWING BE THE RANDOM DISTRIBUTION (User should be able to see only his/her cards)

bot1:DK,SK,DQ,CJ,H10,S9,D8,C7,H6,H5,H4,H3,H2
bot2:HK,SA,DJ,CQ,S10,D9,S8,S7,S6,S5,S4,S3,S2
bot3:HQ,SJ,HJ,CK,C10,C9,C8,H7,C6,C5,C4,C3,C2
player:DA,SQ,HA,CA,D10,H9,H8,D7,D6,D5,D4,D3,D2

calls of players bot1->4, bot2->4, bot3->4, player->4 (IN) cyclic order bot1->bot2->player->bot3->bot1………

```
start s form bot2
Turn1:               Turn 2:               Turn  3:              Turn 4:                  Turn 5:
bot2->SA             bot2->HK              bot2->CQ              player->HA (IN)          player->DA(IN)
player->D2(IN)       player->D3(IN)        player->CA(IN)        bot3->HJ                 bot3->C3
bot3->SJ             bot3->H7              bot3->C2              bot1->H3                 bot1->D8
bot1->S9             bot1->H2              bot1->C7              bot2->S2                 bot2->D9

bot2 wins            bot2 wins             player wins           player wins              player wins

Turn 6:              Turn 7:               Turn 8:               Turn 9:                  Turn 10:
player->SQ(IN)       bot1->DK              bot1->DQ              bot1->CJ                 bot3->HQ
bot3->C4             bot2->DJ              bot2->S4              bot2->S5                 bot1->H4
bot1->SK             player->D4(IN)        player->D5(IN)        player->D6(IN)           bot2->S6
bot2->S3             bot3->C5              bot3->C6              bot3->CK                 player->D7(IN)

bot1 wins            bot1 wins             bot1 wins             bot3 wins                bot3 wins

Turn 11:             Turn 12:              Turn 13:
bot3->C10            bot3->C9              bot3->C8
bot1->H5             bot1->H6              bot1->H10
bot2->S7             bot2->S8              bot2->S10
player->H8(IN)       player->H9(IN)        player->D10(IN)

bot3 wins            bot3 wins             bot3 wins
```

scores:
bot1 = -10*4 (y = 3, x=4)
bot2 = -10*4 (y = 2, x=4)
bot3 = 10*4+(5-4)   (y = 5, x = 4)
player = -10*4 (y = 3, x=4)
player is the winner!!!!!!!

Continue(Y/N): N

Total scores:
bot1 = -10*4 - 10*4 = -80
bot2 = -10*4 -10*4 = -80
bot3 = 10*4+(5-4) -10*4 = 1
player = -10*4 +10*4+2 = 2

player wins the series

==