############### Blackjack Project #####################

#Difficulty Normal 😎: Use all Hints below to complete the project.

#Difficulty Hard 🤔: Use only Hints 1, 2, 3 to complete the project.

#Difficulty Extra Hard 😭: Only use Hints 1 & 2 to complete the project.

#Difficulty Expert 🤯: Only use Hint 1 to complete the project.

############### Our Blackjack House Rules #####################

## The deck is unlimited in size.

## There are no jokers.

## The Jack/Queen/King all count as 10.

## The the Ace can count as 11 or 1.

## Use the following list as the deck of cards:

## cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]

## The cards in the list have equal probability of being drawn.

## Cards are not removed from the deck as they are drawn.

## The computer is the dealer.

##################### Hints #####################

#Hint 1: Go to this website and try out the Blackjack game:

# https://games.washingtonpost.com/games/blackjack/

#Then try out the completed Blackjack project here:

# https://appbrewery.github.io/python-day11-demo/

#Hint 2: Read this breakdown of program requirements:

# http://listmoz.com/view/6h34DJpvJBFVRlZfJvxF

#Then try to create your own flowchart for the program.

#Hint 3: Download and read this flow chart I've created:

# https://drive.google.com/uc?export=download&id=1rDkiHCrhaf9eX7u7yjM1qwSuyEk-rPnt

#Hint 4: Create a deal\_card() function that uses the List below to \*return\* a random card.

#11 is the Ace.

#cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]

#Hint 5: Deal the user and computer 2 cards each using deal\_card() and append().

#user\_cards = []

#computer\_cards = []

#Hint 6: Create a function called calculate\_score() that takes a List of cards as input

#and returns the score.

#Look up the sum() function to help you do this.

#Hint 7: Inside calculate\_score() check for a blackjack (a hand with only 2 cards: ace + 10) and return 0 instead of the actual score. 0 will represent a blackjack in our game.

#Hint 8: Inside calculate\_score() check for an 11 (ace). If the score is already over 21, remove the 11 and replace it with a 1. You might need to look up append() and remove().

#Hint 9: Call calculate\_score(). If the computer or the user has a blackjack (0) or if the user's score is over 21, then the game ends.

#Hint 10: If the game has not ended, ask the user if they want to draw another card. If yes, then use the deal\_card() function to add another card to the user\_cards List. If no, then the game has ended.

#Hint 11: The score will need to be rechecked with every new card drawn and the checks in Hint 9 need to be repeated until the game ends.

#Hint 12: Once the user is done, it's time to let the computer play. The computer should keep drawing cards as long as it has a score less than 17.

#Hint 13: Create a function called compare() and pass in the user\_score and computer\_score. If the computer and user both have the same score, then it's a draw. If the computer has a blackjack (0), then the user loses. If the user has a blackjack (0), then the user wins. If the user\_score is over 21, then the user loses. If the computer\_score is over 21, then the computer loses. If none of the above, then the player with the highest score wins.

#Hint 14: Ask the user if they want to restart the game. If they answer yes, clear the console and start a new game of blackjack and show the logo from art.py.

import random

from art import logo

from replit import clear

def cards\_sum(cards,total):

sum=0

if 'A' in cards:

cards.remove('A')

cards.append('A')

for card in cards:

if card=='J' or card=='Q' or card=='K':

card=10

elif card=='A' and total<=11:

card=11

elif card=='A' and total>11:

card=1

else:

card=card

sum+=card

total=sum

return sum

def total\_without\_A(cards):

sum=0

A\_presence='False'

if 'A' in cards:

cards.remove('A')

A\_presence='True'

for card in cards:

if card=='J' or card=='Q' or card=='K':

card=10

sum+=card

if sum>=21 and A\_presence:

return "False"

else:

return "True"

def check\_next\_game():

next\_game=input("Would you like to play again y or n?")

if next\_game=='y':

continue\_nextgame= "True"

else:

continue\_nextgame= "False"

def presence\_of\_A(cards):

if 'A' in cards:

return True

else:

return False

def check\_user\_total\_is\_21(user\_total):

if user\_total==21:

game\_over=1

print(f"YOU WIN as your cards are {user\_cards} and score is 21 ")

check\_next\_game()

continue\_nextgame='True'

while continue\_nextgame=='True':

game\_over=0

clear()

print(logo)

cards\_list=[2,3,4,5,6,7,8,9,10,'J','Q','K','A']

dealer\_cards=[random.choice(cards\_list)]

user\_cards=[random.choice(cards\_list),random.choice(cards\_list)]

user\_total=0

dealer\_total=0

user\_total = cards\_sum(user\_cards,user\_total)

dealer\_total = cards\_sum(dealer\_cards,dealer\_total)

check\_user\_total\_is\_21(user\_total)

if game\_over==0:

while user\_total<17 or (user\_total>=17 and user\_total<=28 and presence\_of\_A(user\_cards)):

print(f"your cards are {user\_cards} and the sum is {user\_total}")

print(f"dealer's first card is {dealer\_cards[0]}")

next\_chance=input("would you like to hit another card y or n?")

if next\_chance=='y':

user\_cards.append(random.choice(cards\_list))

user\_total=cards\_sum(user\_cards,user\_total)

check\_user\_total\_is\_21(user\_total)

if user\_total>21:

print(f"YOU LOSE, your cards are {user\_cards} and total is {user\_total} and over 21")

game\_over=1

while dealer\_total<17 or (user\_total>=17 and user\_total<=28 and presence\_of\_A(dealer\_cards)):

dealer\_cards.append(random.choice(cards\_list))

dealer\_total=cards\_sum(dealer\_cards,dealer\_total)

if game\_over==0:

if user\_total==dealer\_total:

print(f"IT'S A DRAW, your cards are {user\_cards} and total is {user\_total} and the dealer's cards are {dealer\_cards} and total is {dealer\_total}")

elif user\_total>dealer\_total or dealer\_total>21:

print(f"YOU WIN, your cards are {user\_cards} and total is {user\_total} and the dealer's cards are {dealer\_cards} and total is {dealer\_total}")

else:

print(f"YOU LOSE, your cards are {user\_cards} and total is {user\_total} and the dealer's cards are {dealer\_cards} and total is {dealer\_total}")

check\_next\_game()