**SHARMIDHA SOUNDARARAJAN**

**APPLIED PROGRAMMING - ASSIGNMENT 5**

**PROGRAM 1:**

**class TVShow:**

**def \_\_init\_\_(self,title,genre,rating,noEpisodes,noSeasons):**

**self.title=title**

**self.genre=genre**

**self.rating=rating**

**self.noEpisodes=noEpisodes**

**self.noSeasons=noSeasons**

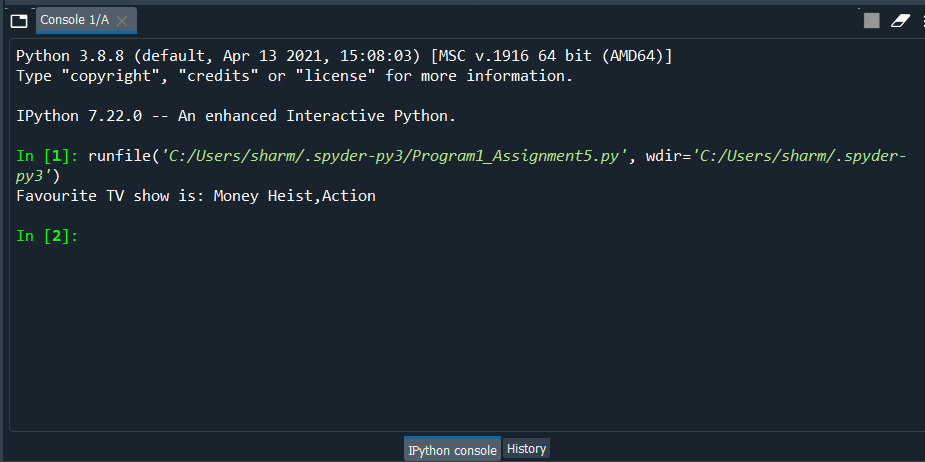
**def \_\_str\_\_(self):**

**return self.title+","+self.genre**

**show=TVShow("Money Heist","Action",4.8,10,5)**

**print("Favourite TV show is:", show)**

**OUTPUT1:**

****

**PROGRAM 2:**

**dict={}**

**letter = input("Enter a scrabble letter: ")**

**while letter!="":**

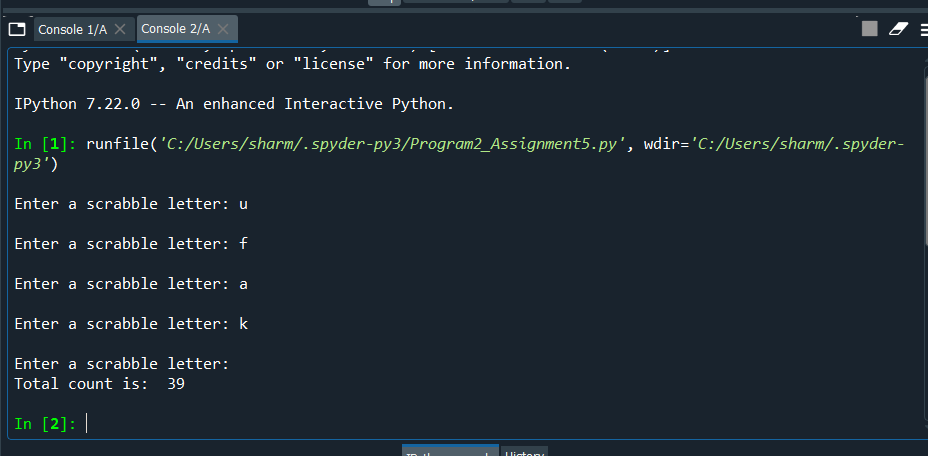
**dict[letter]= ord(letter)-96**

**letter = input("Enter a scrabble letter: ")**

**total\_sum = sum(dict.values())**

**print("Total count is: ", total\_sum)**

**OUTPUT2:**

****

**PROGRAM 3:**

**def hexConvert(number):**

**a=hex(number)**

**value = a.replace("0x", "")**

**return value**

**def info(f):**

**"""The hexConvert function converts an integer given by the user to hexadecimal value and the value is.=="""**

**value = eval(input("Enter an integer: "))**

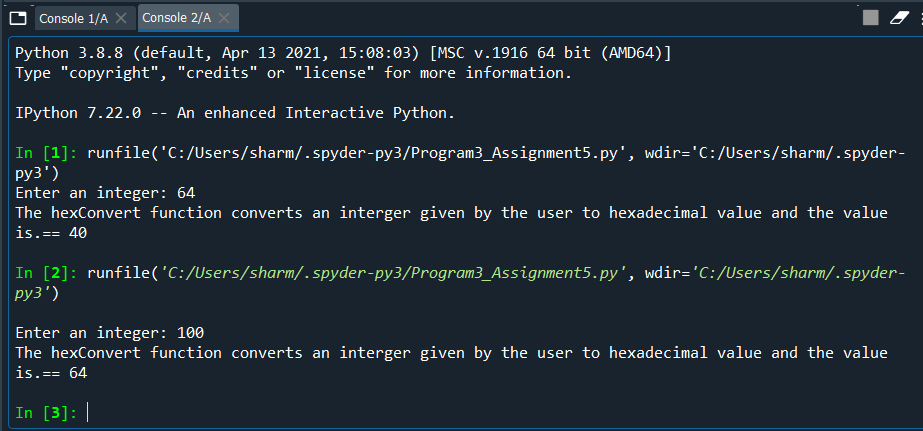
**return f(value)**

**decorator=info(hexConvert)**

**print( info.\_\_doc\_\_,end=" ")**

**print(decorator)**

**OUTPUT3:**

****

**PROGRAM4:**

**OUTPUT 4:**

**import random**

**class PlayingCard:**

**def \_\_init\_\_(self,rank,suit):**

**self.rank = rank**

**self.suit = suit**

**def getRank(self):**

**return self.rank**

**def getSuit(self):**

**return self.suit**

**def value(self):**

**if self.rank>9:**

**return 10**

**else:**

**return self.rank**

**def convertRank(self):**

**if self.rank <12:**

**hexRank ='{0:x}'.format(self.rank)**

**else:**

**hexRank = '{0:x}'.format(self.rank+1)**

**return hexRank**

**def convertSuit(self):**

**suitDictHex = {"h":"b","d":"c","c":"d","s":"a"}**

**return suitDictHex[self.suit]**

**def \_\_str\_\_(self):**

**hexValue = "1f0"+self.convertSuit()+self.convertRank()**

**decValue = int(hexValue,16)**

**return chr(decValue)**

**class Deck:**

**def \_\_init\_\_(self):**

**self.cardDeck = []**

**for r in range(1,14): #ranks**

**for s in 'c','d','h','s': #suits**

**self.cardDeck.append(PlayingCard(r,s))**

**def shuffle(self):**

**random.shuffle(self.cardDeck)**

**def dealCard(self):**

**card = self.cardDeck.pop(0)**

**return card**

**def cardsLeft(self):**

**return len(self.cardDeck)**

**def printDeck(self):**

**for card in self.cardDeck:**

**print(card,end= " ")**

**def sortbySuit(self):**

**self.cardDeck.sort(key=PlayingCard.getSuit)**

**def sortbyRank(self):**

**self.cardDeck.sort(key=PlayingCard.getRank)**

**class Player :**

**def \_\_init\_\_(self, name):**

**self.name = name**

**self.hand = []**

**def drawCard(self, deck):**

**self.hand.append(deck.dealCard())**

**def playCard(self, card):**

**for i in range(len(self.hand)):**

**if self.hand[i].getRank() == card.getRank() and self.hand[i].getSuit() == card.getSuit():**

**self.hand.pop(i)**

**break**

**def getHand(self):**

**return self.hand**

**def sortHand(self):**

**for i in range(len(self.hand)-1):**

**for j in range(len(self.hand)-1):**

**card1 = self.hand[j]**

**card2 = self.hand[j+1]**

**if((card1.getRank() > card2.getRank()) or ((card1.getRank() == card2.getRank()) and (card1.getSuit() > card.getSuit()))):**

**self.hand[j], self.hand[j+1] = self.hand[j+1], self.hand[j]**

**deck = Deck()**

**deck.shuffle()**

**player = Player("Charles")**

**for i in range(6):**

**player.drawCard(deck)**

**print("The cards are: ")**

**hand = player.getHand()**

**for card in hand:**

**print("{}{}".format(card.getSuit(),card.getRank()))**

**player.sortHand()**

**print("Sorted hand is")**

**hand = player.getHand()**

**for card in hand:**

**print("{}{}".format(card.getSuit(),card.getRank()))**

**player.playCard(hand[1]) # remove the 2nd card from hand**

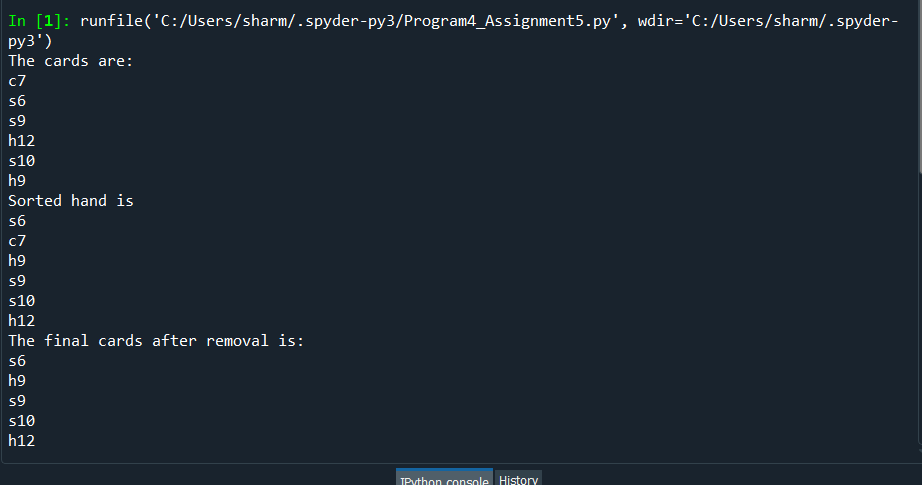
**print("The final cards after removal is:")**

**hand = player.getHand()**

**for card in hand:**

**print("{}{}".format(card.getSuit(),card.getRank()))**

**OUTPUT4:**

****

**PROGRAM 5:**

**import numpy as np**

**def binarySearch(arr, x):**

**low = 0**

**high = len(arr)**

**while (low <= high):**

**m = (low + high) // 2**

**item = arr[m]**

**if (item == x):**

**return m**

**if (item < x):**

**low = m + 1**

**else:**

**high = m - 1**

**return -1**

**dic = open("dict.txt")**

**dictionary = dic.read().split()**

**spellsfile = open("spells.txt")**

**spells = spellsfile.read().split()**

**dictionary.sort()**

**incorrect\_words = 0**

**for spell in spells:**

**result = binarySearch(dictionary, spell)**

**if (result == -1):**

**incorrect\_words+=1**

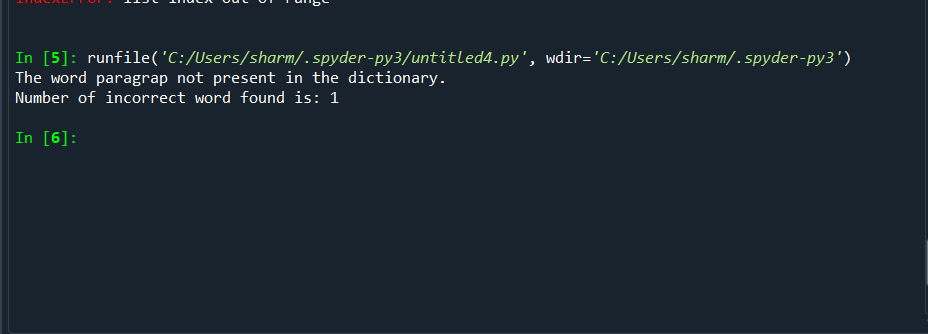
**if (incorrect\_words != 0):**

**print("Number of incorrect word found is: ",incorrect\_words)**

**else:**

**print("All words are present in the dictionary.")**

**OUTPUT 5:**

****