**SHARMIDHA SOUNDARARAJAN**

**APPLIED PROGRAMMING - ASSIGNMENT 4**

**PROGRAM 1:**

**systolic\_bp = eval(input("Enter Systolic blood pressure: "))**

**diastolic\_bp = eval(input("Enter Diastolic blood pressure: "))**

**if systolic\_bp < 120 and diastolic\_bp < 80:**

**print("Blood pressure category is Normal")**

**elif systolic\_bp >= 120 and diastolic\_bp <=129 and diastolic\_bp < 80:**

**print("Blood pressure category is Elevated")**

**elif (systolic\_bp >= 130 and diastolic\_bp <=139) or (diastolic\_bp >= 80 and diastolic\_bp <=89):**

**print("Blood pressure category is Hypertension stage 1")**

**elif (systolic\_bp >= 140 and diastolic\_bp <= 180) or (diastolic\_bp >= 90 and diastolic\_bp <=120):**

**print("Blood pressure category is Hypertension stage 2")**

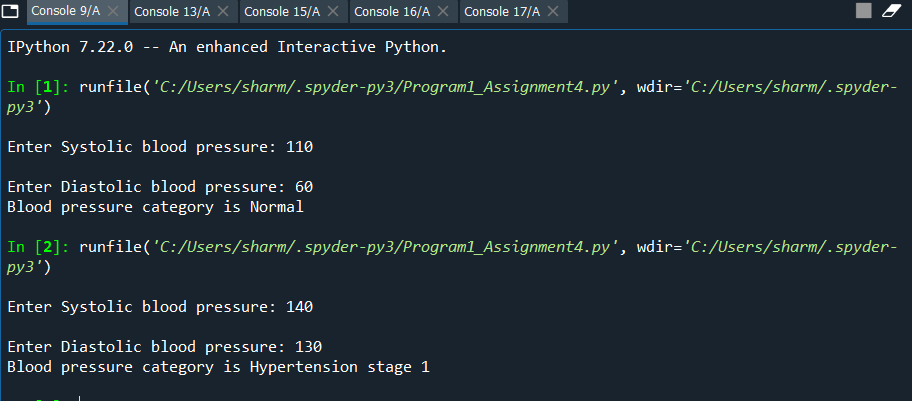
**elif (systolic\_bp > 180) or (diastolic\_bp > 120):**

**print("Blood pressure category is Hypertension crisis")**

**else:**

**print("Enter valid details")**

**OUTPUT1:**

****

**PROGRAM 2:**

**import random**

**from collections import Counter**

**def main():**

**common = 0**

**things = ["Common","Rare","Legendary"]**

**chance = [94.3,5.1,0.6]**

**randomlist = []**

**rare = 0**

**legendary = 0**

**for i in range (1,500):**

**results = random.choices(things,chance)**

**for j in results:**

**if(j == things[1]):**

**rare+=1**

**elif(j == things[2]):**

**legendary+=1**

**# check if Rare card is not drawn after 9 attempts**

**if i<9:**

**if results==things[1]:**

**break**

**if (rare==0) and (i == 9):**

**results = things[1]**

**rare+=1**

**#If a Legendary card is not drawn after 75 attempts, prob of drawing legendary, increases to 30%.**

**if(i>=76) and (legendary==0):**

**for j in results:**

**chance[2] = 30**

**chance[0] = 64.9 #decrease the probability of common to have total probabilty to 1**

**#If a Legendary card is drawn after 75 attempts, put the probabilty back to normal**

**elif(i>=76) and (legendary!=0):**

**if(j == 'Legendary'):**

**chance[2] = 0.6**

**chance[0] = 94.3**

**# check if Legendary card is not drawn after 89 attempts**

**if (legendary==0) and (i == 89):**

**results = 'Legendary'**

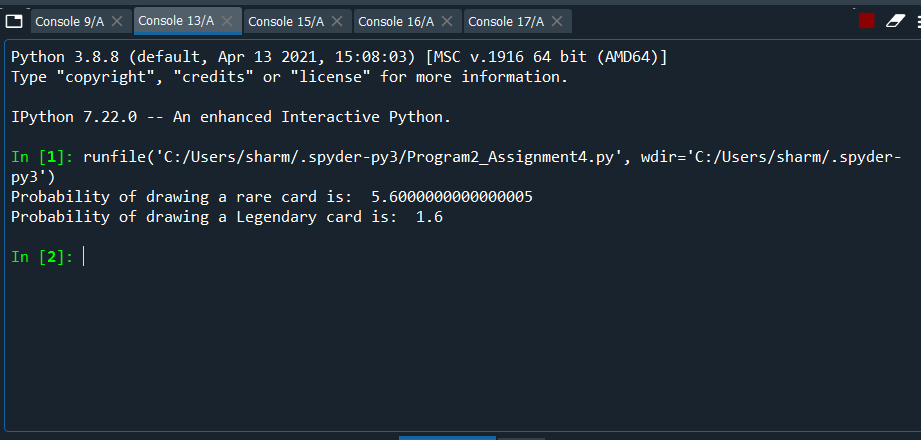
**legendary+=1**

**print("Probability of drawing a rare card is: ", (rare/500)\*100)**

**print("Probability of drawing a Legendary card is: ", (legendary/500)\*100)**

**main()**

**OUTPUT2**

****

**PROGRAM 3:**

**import numpy as np**

**np.random.seed(10)**

**shared\_birthday = 0**

**probabilities = []**

**size = 30**

**for i in range(0,100):**

**# generate random dates to 30 memebers**

**dates = np.random.choice(range(1,366), size=30)**

**# check if there are duplicates**

**if len(np.unique(dates)) != len(dates):**

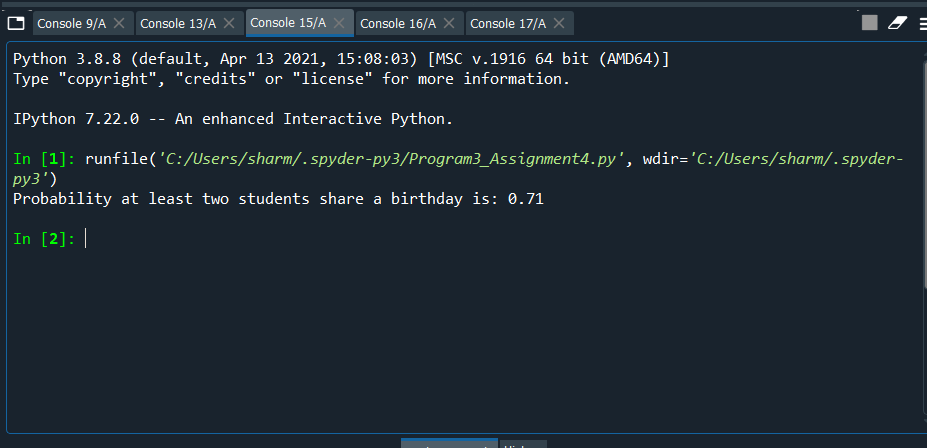
**shared\_birthday+=1**

**probability = shared\_birthday/100**

**probabilities.append(probability)**

**print("Probability at least two students share a birthday is:" , probability)**

**OUTPUT3:**

****

**PROGRAM4:**

**import numpy as np**

**import random**

**np.random.seed(10)**

**def generate\_random():**

**#generate random numbers for a pair of dice and add them**

**dice1 = np.random.randint(1,6)**

**dice2 = np.random.randint(1,6)**

**dice = dice1+dice2**

**return dice**

**def main():**

**lose = 0**

**win=0**

**for i in range(0,10000):**

**roll = generate\_random()**

**if roll==2 or roll==3 or roll==12:**

**lose+=1**

**elif roll==7 or roll==11:**

**win+=1**

**else:**

**re\_roll=0**

**initial\_roll=roll**

**#re-roll the dice**

**while (re\_roll!=initial\_roll):**

**re\_roll = generate\_random()**

**#check if the dice value is equal to 7**

**if re\_roll == 7:**

**lose+=1**

**break**

**if re\_roll==initial\_roll:**

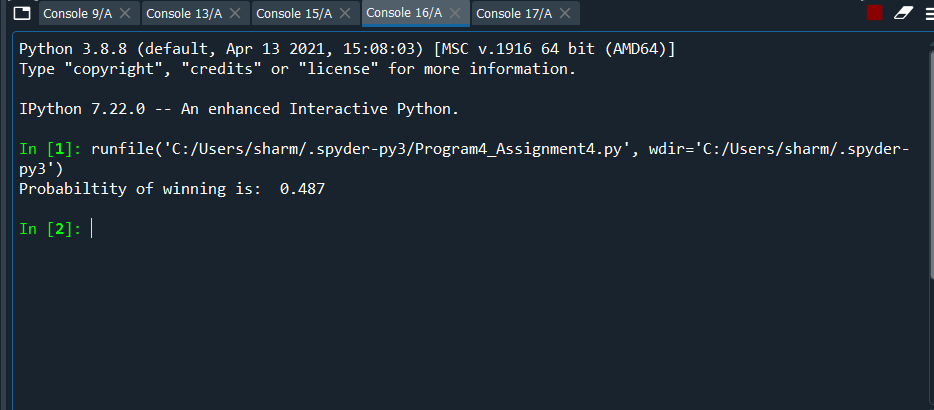
**win+=1**

**probabiltity = win/10000**

**print("Probabiltity of winning is: ",probabiltity)**

**main()**

**OUTPUT 4:**

****

**PROGRAM5:**

**import random**

**import math**

**import numpy as np**

**def main():**

**time = 0**

**for i in range(1,1000):**

**x = 0**

**y = 0**

**distance=0**

**while(distance <= 1):**

**result = ['N','S','E','W']**

**direction = random.choices(result) #randomly generate values in 4 directions**

**for j in direction:**

**if(j == 'N'):**

**y += 0.6**

**elif(j == 'S'):**

**y -= 0.6**

**elif(j == 'E'):**

**x += 0.6**

**else:**

**x -= 0.6**

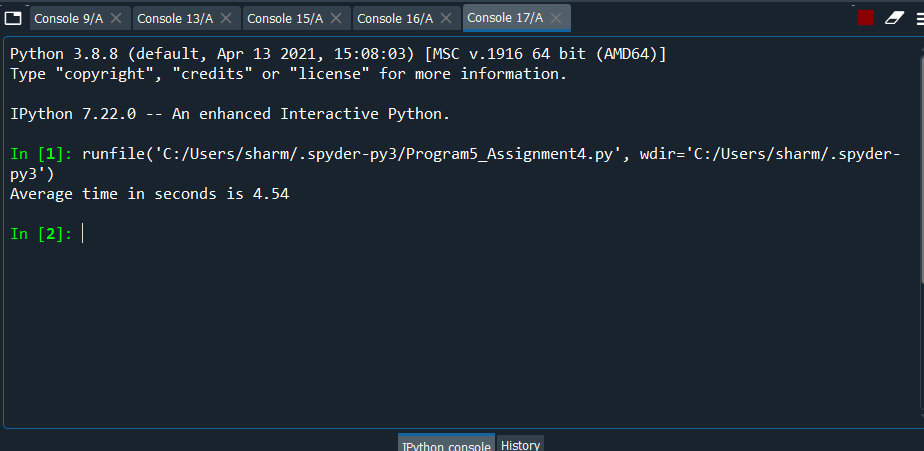
**distance = math.sqrt((x)\*\*2 + (y)\*\*2)**

**time += 1**

**print("Average time in seconds is", round(time/1000, 2))**

**main()**

**OUTPUT 5:**

****