**Q1.Write a python program which has two functions to generate the frequency matrix for the characters present in the given string. First function should take the string input and return the dictionary with the keys being the characters in the string and the values for those keys being the frequency of that character in the string.**

**Code:**

**#Program to display frequency of letters in a string**

**dict1={}**

**def frequency(str1):**

**for i in range(len(str1)):**

**dict1[str1[i]]=str1.count(str1[i])**

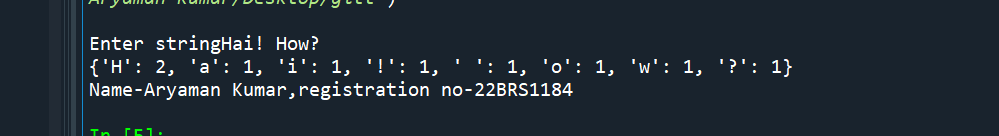
**return dict1**

**str1=input("Enter string")**

**print(frequency(str1))**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**Q2.Write a Python code using functions to print the number of uppercase, lowercase and special characters available in the given string. The function freqcounter() should take the string input as one of its parameter and return the count of uppercase, lowercase and the special character count to the driver program. As these are to be returned as a dictionary from the function.**

**#Program to store number of uppercase,lowercase and special characters in a string**

**dict1={}**

**def freqcounter(str1):**

**Uc=Uppercase(str1)**

**lc=lowercase(str1)**

**dig=digit(str1)**

**dict1["Uppercase"]=Uc**

**dict1["lowercase"]=lc**

**dict1["Special character"]=len(str1)-(dig+Uc+lc)**

**return dict1**

**def Uppercase(str1):**

**c1=0**

**for i in range(len(str1)):**

**if(str1[i].isupper()==True):**

**c1+=1**

**return c1**

**def lowercase(str1):**

**c2=0**

**for i in range(len(str1)):**

**if(str1[i].islower()==True):**

**c2+=1**

**return c2**

**def digit(str1):**

**c3=0**

**for i in range(len(str1)):**

**if(str1[i].isdigit()==True):**

**c3+=1**

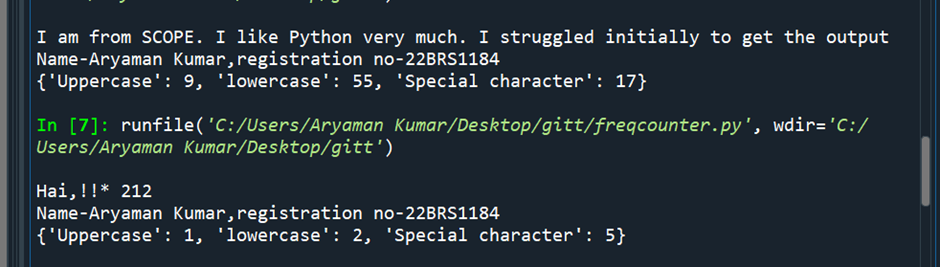
**return c3**

**str1=input()**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**print(freqcounter(str1))**

**Output:**

****

**Develop a Python code with two functions: consonants(word) which returns the number of consonants in a word when called. vowels(word) which returns the number of vowels in a word when called. Spaces and special character do not count as consonants nor vowels. Here the names of the functions are consonants(word) and vowels(word). word is the parameter that you pass for that function.**

**Code:**

**Output:**

**Create a function to test if a string is a valid pincode or not. A valid pin has: Exactly 6 characters. Only numerical characters (0-9). It should not start with zero. First digit of the pin code must be from 1 to 9. No whitespace.**

**#Program to check if pincode is valid or not**

**import re**

**pin=input()**

**x=re.search("^[1-9].{4}[0-9]$",pin)**

**print(x)**

**if x:**

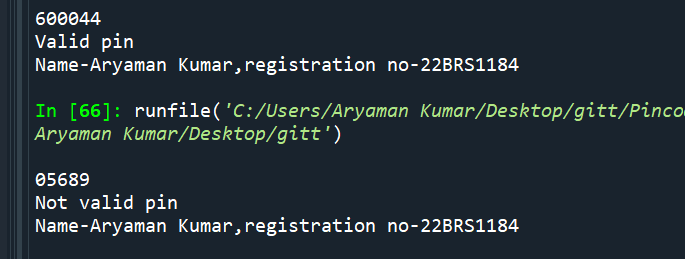
**print("Valid pin")**

**else:**

**print("Not valid pin")**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**5. Encrypt a given message by “rotating” each letter by a fixed number of places. To rotate a letter means to shift it through the alphabet, wrapping around to the beginning if necessary, so ‘A’ rotated by 3 is ‘D’ and ‘Z’ rotated by 1 is ‘A’. Write a function called rotate\_word that takes a string and an integer as parameters, and returns a new string that contains the letters from the original string rotated by the given amount. Check whether the output string has the vowels in the beginning and at the end of the string. If the condition satisfies, display the message “Happy Cool String” else display “Happy Hot String”.**

**Code:**

**#Program to encrypt a string by adding specified number to ASCII value**

**def rotate\_word(str1,n):**

**rt=""**

**b=0**

**for i in range(len(str1)):**

**b=ord(str1[i])+n**

**rt+=chr(b)**

**return rt**

**a=input()**

**n1=int(input())**

**c=rotate\_word(a,n1)**

**print(c)**

**if c[0] in ('a','e','i','o','u','A','E','I','O','U') and c[len(c)-1] in('a','e','i','o','u','A','E','I','O','U'):**

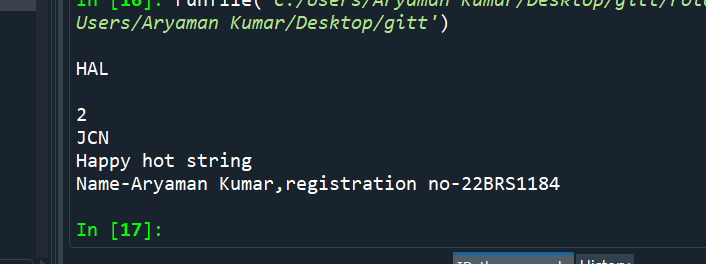
**print("Happy cool string") #To check if first and last letter are vowels**

**else:**

**print("Happy hot string")**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**6. Write a Python program to check whether the date of birth that is given as input is in the format YYYY-MMM-DD. If it is available leave it as such displaying the message, “Correct Format”. Else, Convert the given input (DD-MM-YYYY) into the required format and display the result along with the message “Format Corrected”. Use regular expression for this program to check the format of the given string.**

**Code:**

**Output:**

**7.Given two strings, s1 (all Upper Case) and s2 (All lowerCase), create a mixed String Instructions: Create a third-string made of the first char of s1 then the last char of s2, Next, the second char of s1 and second last char of s2, and so on. Any leftover chars go at the end of the result. Check whether the output string follows the pattern of Upper and lower case alphabets being available in the alternate position. If not in the expected pattern, Please display "Does Not Follow Any Pattern" If it follows the pattern, then display, "Follows Pattern".**

**8.Write a program that takes your full name as input and displays the abbreviations of the first and middle names except the last name which is displayed as it is. For example, if your name is Robert Brett Roser, then the output should be R.B.Roser.**

**Code:** **#Program to display only the Initials and lastname of a given name**

**str1=input()**

**list1=str1.split(" ")**

**str2=""**

**for i in range(0,len(list1)-1):**

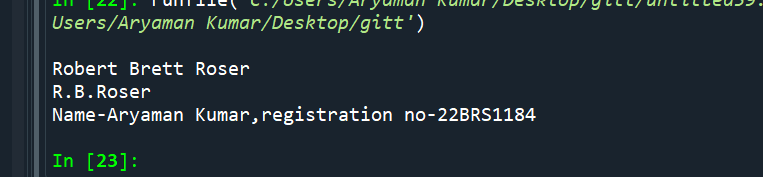
**str2=str2+list1[i][0]+"."**

**str2=str2+list1[len(list1)-1]**

**print(str2)**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**9. Get N names in a list and prints the shortest names in the list, one on each line. There is more than one shortest name, so both need to be printed.**

**Code:**

**#Program to print shortest name in list**

**list1=[]**

**N=int(input())**

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

**a=input()**

**list1.append(a)**

**list1.sort(key=len)**

**print(list1[0])**

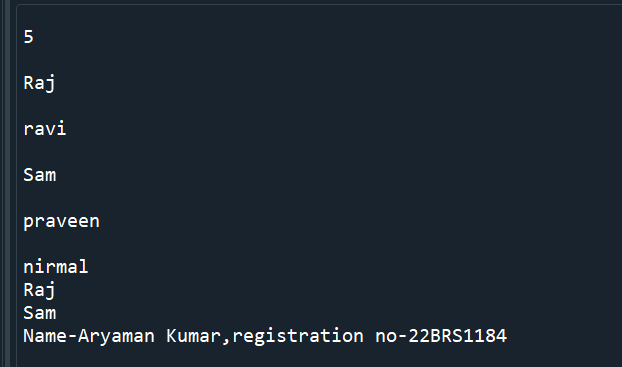
**for i in list1:**

**if len(i)==len(list1[0]) and i!=list1[0]:**

**print(i)**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**10 From the given string, find the string length and check whether the string is palindrome or not.**

**Code:** **def palindrome(str1):**

**str1=str1.lower()**

**if(str1==str1[::-1]):**

**return True**

**else:**

**return False**

**str1=input()**

**print(len(str1))**

**if((palindrome(str1)==True)):**

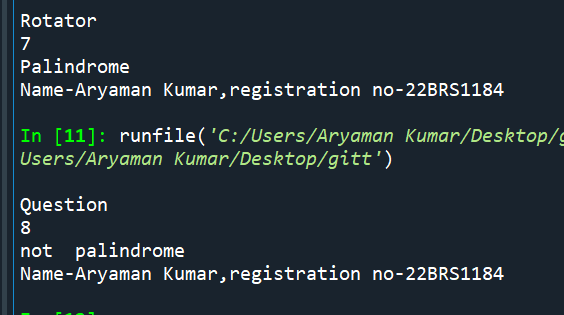
**print("Palindrome")**

**else:**

**print("not palindrome")**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**11.Get a Python list of N integer values. Find the presence of ODD and EVEN numbers in the list. If the ODD number occurs at the ODD position replace the ODD number with 'odd'. If the EVEN number occurs at the EVEN position replace the EVEN number with 'even'.**

**Find the number of odd places that were changed and also the number of even places that were changed**

**Prepare a new list for the odd and even positions.**

**Code:**

**list1=[]**

**odd=0**

**even=0**

**oddposlist=[]**

**evenposlist=[]**

**N=int(input())**

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

**a=int(input())**

**list1.append(a)**

**for i in range(len(list1)):**

**if((i+1)%2!=0 and list1[i]%2!=0):**

**odd+=1**

**oddposlist.append(i)**

**list1[i]='odd'**

**elif((i+1)%2==0 and list1[i]%2==0):**

**even+=1**

**evenposlist.append(i)**

**list1[i]='even'**

**print(list1)**

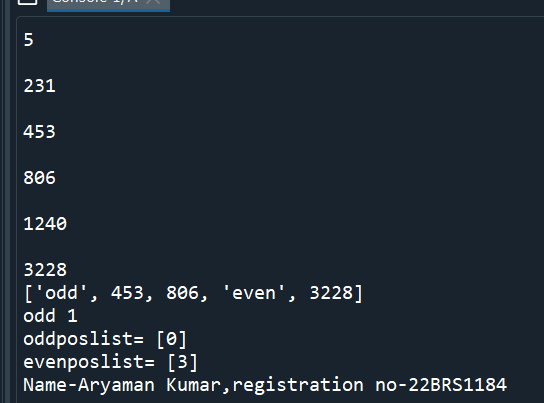
**print("odd",odd)**

**print("oddposlist=",oddposlist)**

**print("evenposlist=",evenposlist)**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**12. Get the following states and their number of districts as key and value pairs in a dictionary. Example: {'GOA':2, 'LADAKH':2, 'PUDUCHERRY':4, 'SIKKIM':4, 'DELHI':11, 'MEGHALAYA':11, 'MIZORAM':11, 'ANDHRA PRADESH':13, 'UTTARAKHAND':13, 'PUNJAB':23, 'WEST BENGAL':23, 'ASSAM':33, 'GUJARAT':33, 'RAJASTHAN':33, 'TELANGANA':33} For every entered state, - find its number of districts - count the number of states that has the same count of districts - Form a new dictionary from the above-founded number of districts as key and the list of states that have the same number of districts as its value.**

**13. Get a list of N elements. Split and store it into two different list based on the middle element. Have only the even number of elements in the list.**

**Code:**

**#Program to split a list**

**N=int(input())**

**if(N%2==0):**

**list1=[]**

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

**n=int(input())**

**list1.append(n)**

**list2=list1[0:N//2]**

**list3=list1[N//2:len(list1)]**

**print("list1:")**

**for i in list2:**

**print(i)**

**print("list2:")**

**for i in list3:**

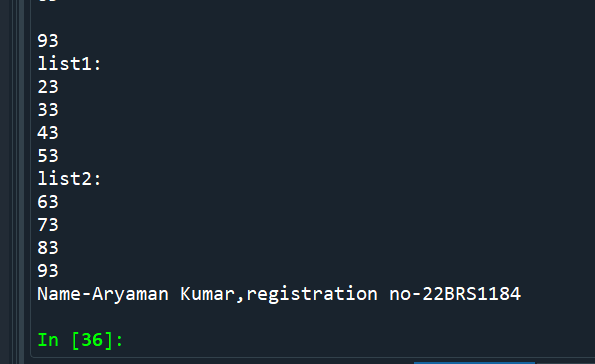
**print(i)**

**else:**

**print("Even number of elements required")**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**14. Execute a python program for the below task. Get a number (between 1 and 10) from the user and create a dictionary as following key-value pair (key and a list as value), if the entered number is 3, the output is: {1:[1], 2:[1,2], 3:[1,2,3]}.**

**Code:**

**n=int(input())**

**dict1={}**

**list1=[]**

**if(n>0 and n<=10):**

**for i in range(1,n+1):**

**for j in range(1,i+1):**

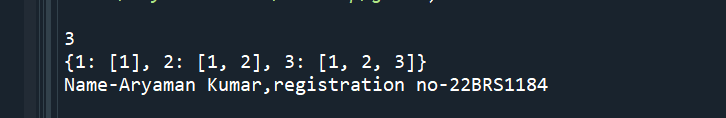
**list1.append(j)**

**dict1[i]=list1**

**list1=[]**

**print(dict1)**

**Output:**

****

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**15. IPS for Dictionary: Execute a Python program to check whether a given key already exists in a dictionary.**

**Code:**

**dict1={}**

**n=int(input())**

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

**k=int(input())**

**v=int(input())**

**dict1[k]=v**

**s=int(input())**

**if s in dict1:**

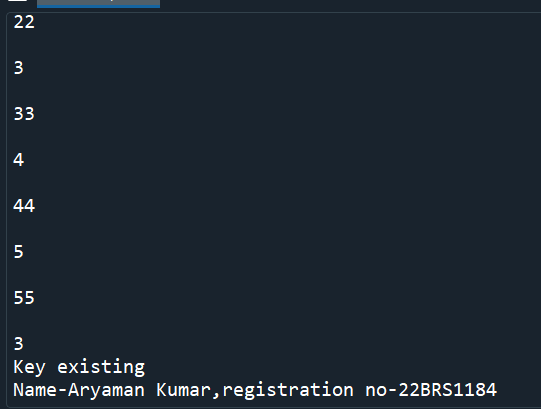
**print("Key existing")**

**else:**

**print("Key not existing")**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**16. IPS for Tuples: Consider a tuple as T = (1, 3, 2, 4, 6, 5). Write a program to store numbers present at odd index into a new tuple.**

**Code:**

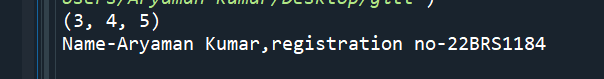
**T=(1, 3, 2, 4, 6, 5)**

**T2=T[1:len(T):2]**

**print(T2)**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**17. Write a code to count the numbers that are divisible by 2 and 3. Accept numbers as input one below the other from the user till -100 is pressed Display the DivisibleCount and NotDivisibleCount values as result .**

**Code:**

**Divisiblecount=0**

**Nondivisiblecount=0**

**while(True):**

**n=int(input())**

**if(n==-100):**

**print("Divisible count is ",Divisiblecount)**

**print("Nondivisible count is",Nondivisiblecount)**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

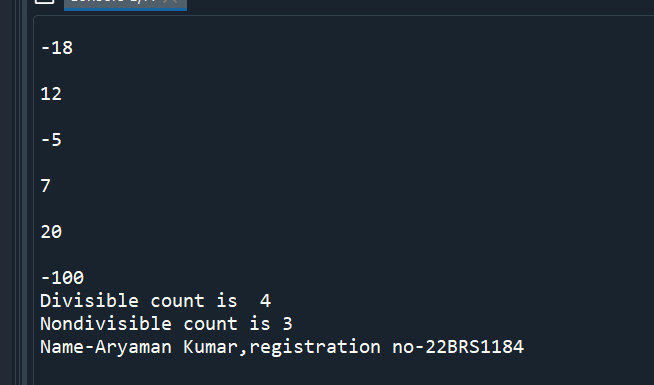
**if(n%2==0 and n%3==0):**

**Divisiblecount+=1**

**else:**

**Nondivisiblecount+=1**

**Output:**

****

**18. Given an array of integers, return True if 6 appears as either the first or last element in the array. The array will be length 1 or more.**

**Code:**

**n=eval(input())**

**if (n[0]==6 or n[len(n)-1]==6 and len(n)>1):**

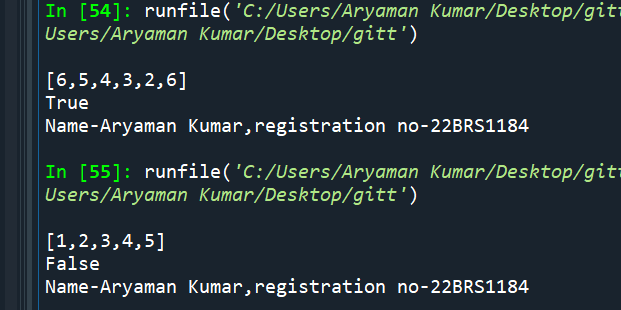
**print( True)**

**else:**

**print(False)**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****

**19. Given a list of marks secured in CSE1001 by the students, write a Python code to determine the class average. Print only two decimal digits in average.**

**Code:** **#Program to print average of list with two zeroes**

**n=int(input())**

**list1=[]**

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

**a=int(input())**

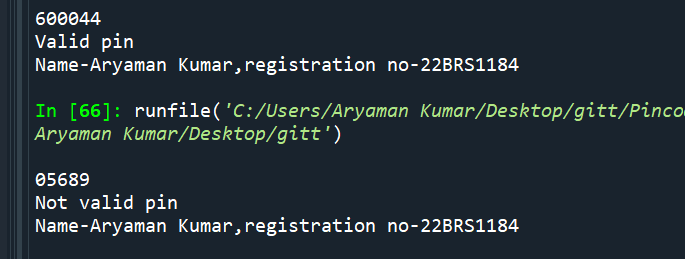
**list1.append(a)**

**avg=float(sum(list1)/len(list1))**

**print("%.2f" % avg)**

**print("Name-Aryaman Kumar,registration no-22BRS1184")**

**Output:**

****