***PYTHON***

***Lab Copy***

**Name:Shantanu Swaraj**

**Sap id: 500097021**

**Batch: b-41**

**Roll no.:**

**INDEX**

1. **Experiment 1**
2. **Experiment 2**
3. **Experiment 3**
4. **Experiment 4**
5. **Experiment 5**
6. **Experiment 6**

**EXPERIMENT . 1**

**Question 1.1:**

**Given an integer n, perform the following conditional actions:  
 If n is odd, print Weird.  
 If n is even and in the inclusive range of 2 to 5, print Not Weird.  
 If n is even and in the inclusive range of 6 to 20, print Weird.  
 If n is even and greater than 20, print Not Weird.**

* **CODE:**

**#question no 1.1**

**n=int(input("enter value for n:"))**

**if n%2==1:**

**print("wierd")**

**elif n%2==0 and 2<=n<=5:**

**print("not wierd")**

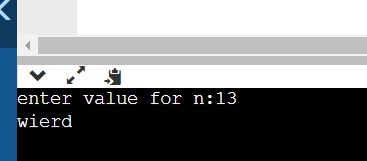
**elif n%2==0 and 6<=n<=20:**

**print("wierd")**

**else :**

**print("not wierd")**

* **Output:**



**Question 1.2:**

**WAP to read an integer ‘n’ from STDIN. For all non-negative integers i<n, print i2 on a separate line?**

* **Code:**

**#question no.1.2**

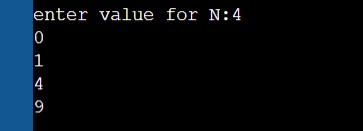
**#STDIN stands for standard input**

**N=int(input("enter value for N:"))**

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

**print(i\*i)**

* **Output:**



**Question 1.3:**

**WAP to read an integer from STDIN. Without using any string methods, print the following on a single line:**

* **Code:**

**#question no. 1.3**

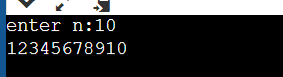
**if \_\_name\_\_=='\_\_main\_\_':**

**n=int(input("enter n:"))**

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

**print(i+1,end="")**

* **Output:**

****

**Question 2&3.1:**

**WAP to read the record of n students in a dictionary containing key/value pairs of name: [marks]. Print the average of the marks obtained by the particular student correct to 2 decimal places.**

* **Code:**

**n=int(input("enter value for n:"))**

**student\_mark={}**

**for \_ in range(n):**

**name,\*line=input().split()**

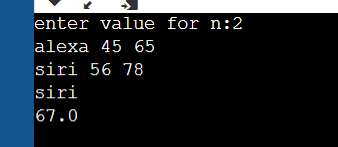
**scores=list(map(float,line))**

**student\_mark[name]=scores**

**query\_name=input()**

**print("{0:.1f}".format(round(sum(student\_mark[query\_name])/len(student\_mark[query\_name]),2)))**

* **Output:**

****

**Question 2&3 .2:**

**WAP to input a list of scores for N students in a list data type. Find the score of the runner-up and print the output.**

* **Code:**

**#question 2&3 .2**

**i=int(input("enter value for i:"))**

**list=list(map(int,input().strip()))[:i]**

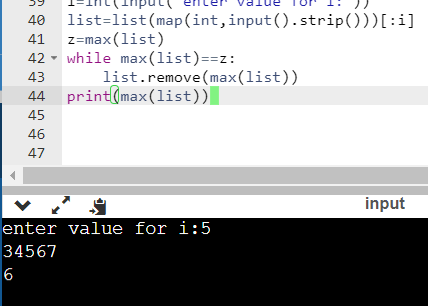
**z=max(list)**

**while max(list)==z:**

**list.remove(max(list))**

**print(max(list))**

* **Output:**

****

**Question 2&3 .3:**

**Rupal has a huge collection of country stamps. She decided to count the total number of distinct country stamps in her collection. She asked for your help. You pick the stamps one by one from a stack of country stamps. Find the total number of distinct country stamps using a suitable data type.**

* **Code:**

**#question 2&3 .3:**

**n=int(input("enter value for n:"))**

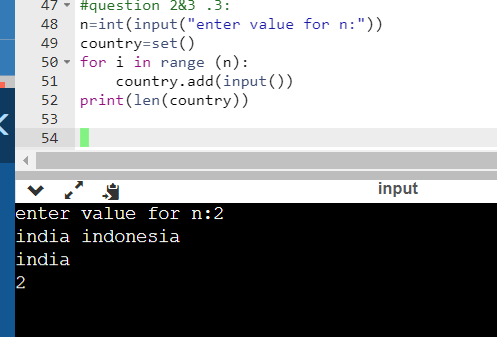
**country=set()**

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

**country.add(input())**

**print(len(country))**

* **Output:**



**Question no 4.1:**

**WAP to enter a string and a substring. You have to print the number of times that the substring occurs in a given string. String traversal will take place from left to right, not from right to left.**

* **Code:**

**#question no. 4.1**

**def count\_substring(string,sub\_string):**

**count=0**

**for i in range(len(string)-len(sub\_string)+1):**

**if(string[i:i+len(sub\_string)]==sub\_string):**

**count+=1**

**return count**

**if \_\_name\_\_=='\_\_main\_\_':**

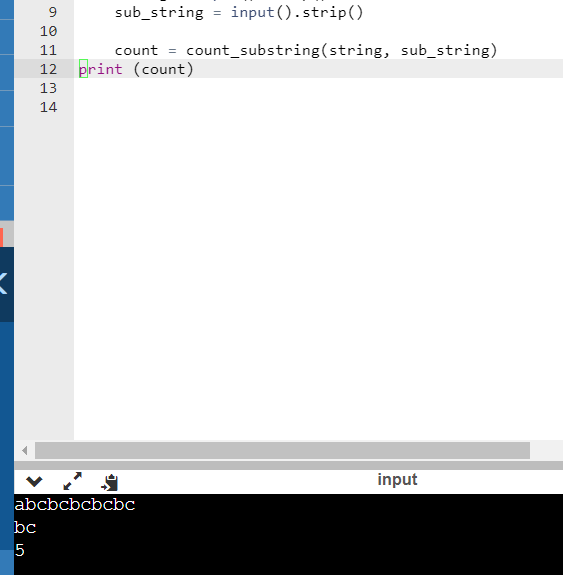
**string=input().strip()**

**sub\_string=input().strip()**

**count=count\_substring(string,sub\_string)**

**print(count)**

* **Output:**



**Question 4.2:**

**WAP to input the first name. middle and last name of a person. Your task is to print the initials of the first and middle name separated by a dot(.)  
The last name should be followed by a dot and a space where the first letter is capital.**

* **Code:**

**#question no. 4.2**

**name=input("enter your name:")**

**m = name.split()**

**x=m[0]**

**y=m[1]**

**z=m[2]**

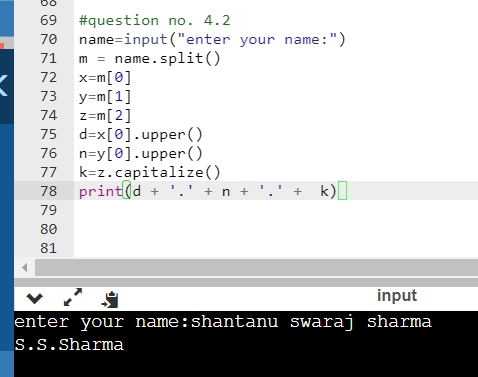
**d=x[0].upper()**

**n=y[0].upper()**

**k=z.capitalize()**

**print(d + '.' + n + '.' + k)**

**output:**



**Question 4.3:**

**Given a string containing both upper and lower case alphabets. Write a python program to count the number of occurrences of each alphabet (case sensitive) and display the same.**

* **Code:**

**string=input("enter the string:")**

**newstr=list(string)**

**newlist=[]**

**for j in newstr:**

**if j not in newlist:**

**newlist.append(j)**

**count=0**

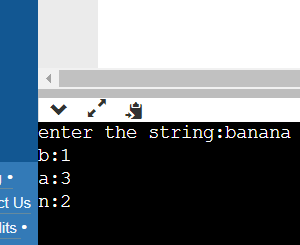
**for i in range(len(newstr)):**

**if j==newstr[i]:**

**count+=1**

**print("{}:{}".format(j,count))**

* **Output:**



**Question no. 5&6:**

**5&6.1:** **Using functions, re-write and execute python program to:**

**1.Add natural number upto n where n is taken as an input from user.**

**2.Print Fibonacci series till nth term (Take input from user).**

**-> code:**

**# part 1**

**num=int(input("enter the value of n:"))**

**hold = num**

**sum = 0**

**if num <= 0:**

**print("enter a whole positive number")**

**else:**

**while num > 0:**

**sum=sum+num**

**num=num-1**

**print("sum of first",hold,"natural number is:",sum)**

**# part 2**

**n=int(input("number of terms:"))**

**n1,n2=0,1**

**count = 0**

**if n <= 0:**

**print("enter a positive integer")**

**elif n==1:**

**print("fibonacci sequence upto",n,":")**

**print(n1)**

**else:**

**print("fibonacci sequence:")**

**while count < n:**

**print(n1)**

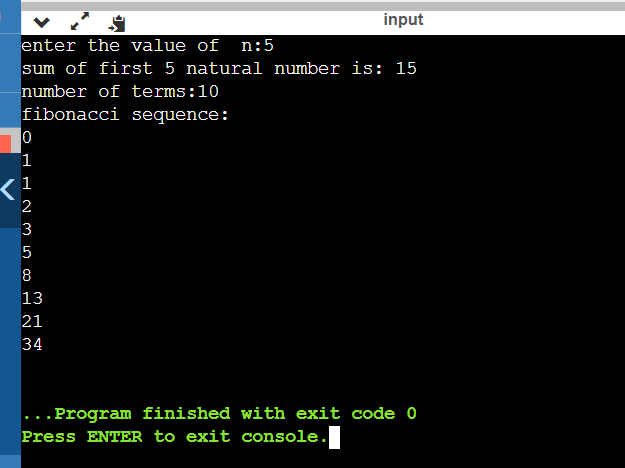
**nth = n1 + n2**

**n1=n2**

**n2=nth**

**count += 1**

* Output:



**Question 5&6:**

**5&6.3:** **Write a Python program to find the maximum and minimum values in a given list of tuples using lambda function.**

* **Code:**

**def max\_min\_list\_tuple(class\_students):**

**return\_max=max(class\_students,key=lambda item: item[1])[1]**

**return\_min=min(class\_students,key=lambda item:item[1])[1]**

**return return\_max,return\_min**

**class\_students=[('I',50),('II',55),('III',60),('IV',65)]**

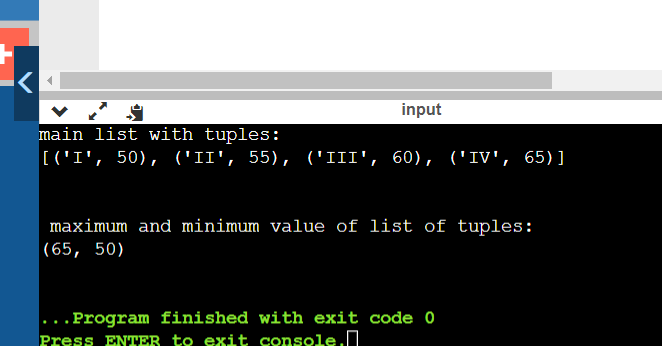
**print("main list with tuples:")**

**print(class\_students)**

**print("\n\n maximum and minimum value of list of tuples: ")**

**print(max\_min\_list\_tuple(class\_students))**

* **Output:**



**Question 5&6:**

**5&6.2:** **At an airport, a traveller is allowed entry into the flight only if he clears the following checks:**

**1. Baggage Check**

**2. Immigration Check**

**3. Security Check**

**The logic for the check methods are given below:**

**check\_baggage (baggage\_weight)**

**returns True if baggage\_weight is greater than or equal to 0 and less than or equal to 40. Otherwise returns False.**

**check\_immigration (expiry\_year)**

**returns True if expiry\_year is greater than or equal to 2030 and less than or equal to 2050. Otherwise returns False.**

**check\_security(noc\_status)**

**returns True if noc\_status is 'valid' or 'VALID', for all other values return False.**

**traveller ()**

**Initialize the traveler Id and traveler name and invoke the functions check\_baggage(), check\_immigration() and check\_security() by passing required arguments.**

**If all values of check\_baggage(), check\_immigration() and check\_security() are true,**

**- display traveler\_id and traveler\_name**

**- display "Allow Traveler to fly!"**

**Otherwise,**

**- display traveler\_id and traveler\_name**

**- display "Detain Traveler for Re-checking!**

**Invoke the traveler() function. Modify the values of different variables in traveler() function and observe the output.**

* **Code:**

**def check\_baggage(baggage\_weight):**

**if baggage\_weight>0 and baggage\_weight<40:**

**print("true")**

**else:**

**print("false")**

**def check\_immigration(expiry\_date):**

**if expiry\_date >= 2030 and expiry\_date <=2050:**

**print("true")**

**else:**

**print("false")**

**def noc\_status(m):**

**if m=="valid":**

**print("true")**

**else:**

**print("false")**

**def travellor():**

**name=input("enter trveller's full name:")**

**ID=input("enter traveller's ID:")**

**baggage\_weight=int(input("enter baggage\_weight in kilogram:"))**

**expiry\_date=int(input("enter expiry date of password:"))**

**noc\_status10=input("specify if your noc status is valid or invalid:")**

**check\_baggage(baggage\_weight)**

**check\_immigration(expiry\_date)**

**noc\_status(noc\_status10)**

**if check\_baggage=="true" and check\_immigration=="true" and noc\_status10=="valid":**

**print(name+" "+ID)**

**print("allow travellor to fly, have a happy journey")**

**else:**

**print(name+" "+ID)**

**print("detain travellor for re-checking")**

**print("\nHello, DEAR CUSTOMERS\nWelcome to RISE.Please help us to check your details")**

**travellor()**

* **Output:**

