**Computer Science and Engineering Department**

**Artificial Intelligence (UCS 521)**

**Lab Assignment-1**

1. **A class with 10 students wants to produce some information from the results of the four standard tests in Maths, Science, English and IT. Each test is out of 100 marks. The information output should be the highest, lowest and average mark for each test and the highest, lowest and average mark overall. Write a program in Python to complete this task.**

**CODE:**

print("Enter marks of students in Maths : ",end=" ")

M = list(int(i) for i in input().split())

print(M)

print("Enter marks of students in Science : ",end = " ")

S = list(int(i) for i in input().split())

print(S)

print("Enter marks of students in English : ",end=" ")

E = list(int(i) for i in input().split())

print(E)

print("Enter marks of students in IT : ",end=" ")

I = list(int(i) for i in input().split())

print(I)

print("\n")

print("Maximum Marks in Maths = ",max(M))

print("Minimum Marks in Maths = ",min(M))

print("Average Marks in Maths = ",sum(M)/len(M))

print("\n")

print("Maximum Marks in Science = ",max(S))

print("Minimum Marks in Science = ",min(S))

print("Average Marks in Science = ",sum(S)/len(S))

print("\n")

print("Maximum Marks in English = ",max(E))

print("Minimum Marks in English = ",min(E))

print("Average Marks in English = ",sum(E)/len(E))

print("\n")

print("Maximum Marks in IT = ",max(I))

print("Minimum Marks in IT = ",min(I))

print("Average Marks in IT = ",sum(I)/len(I))

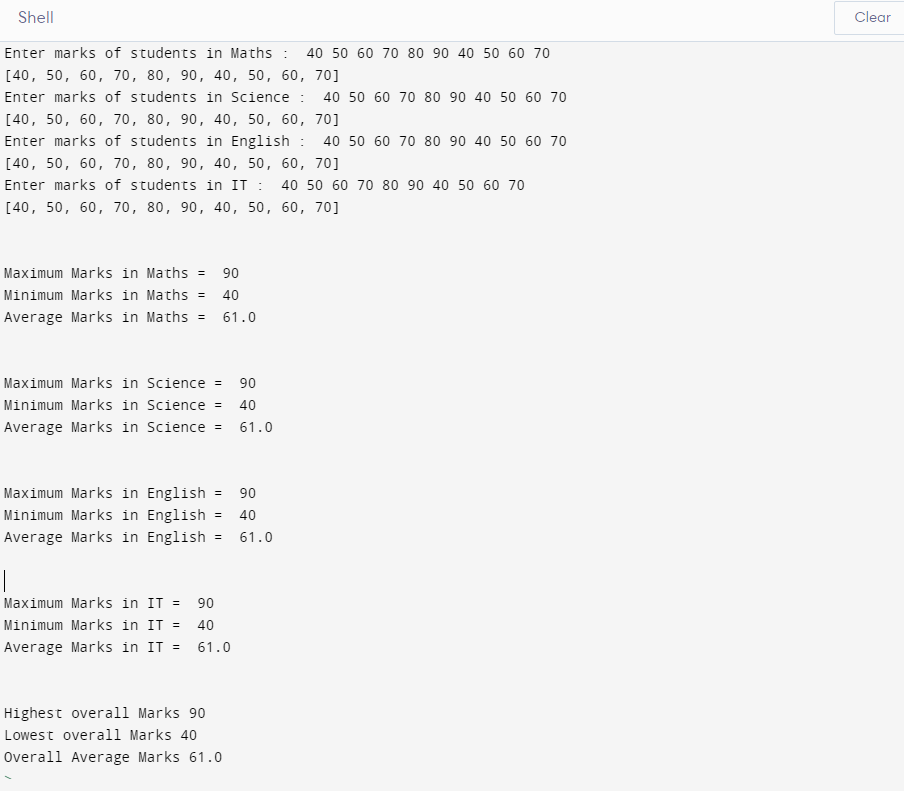
print("\n")

print("Highest overall Marks",max(max(M),max(max(S),max(max(E),max(I)))))

print("Lowest overall Marks",min(min(M),min(min(S),min(min(E),min(I)))))

print("Overall Average Marks",(sum(E)+sum(M)+sum(S)+sum(I))/40)

**OUTPUT:**



1. **Write a Python Program to input basic salary of an employee and calculate its Gross salary according to following: Basic Salary <= 10000 : HRA = 20%, DA = 80% Basic Salary <= 20000 : HRA = 25%, DA = 90% Basic Salary > 20000 : HRA = 30%, DA = 95%.**

**CODE:**

print("Enter Basic Salary = ",end=" ")

basic\_salary=int(input())

print(basic\_salary)

if basic\_salary <= 10000 :

hra = 0.20\*basic\_salary

da = 0.80\*basic\_salary

elif basic\_salary <= 20000 :

hra = 0.25\*basic\_salary

da = 0.90\*basic\_salary

else:

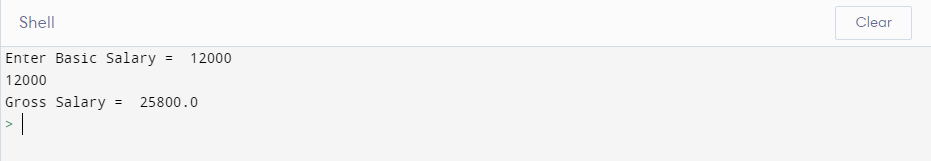
hra = 0.30\*basic\_salary

da = 0.95\*basic\_salary

gross\_salary = basic\_salary + hra + da

print("Gross Salary = ",gross\_salary)

**OUTPUT:**



1. **Write a Python program to check the validity of password input by users. Validation:**

** At least 1 letter between [a-z] and 1 letter between [A-Z].**

** At least 1 number between [0-9].**

** At least 1 character from [$#@].**

** Minimum length 6 characters.**

** Maximum length 16 characters.**

**CODE:**

def isValid(str):

if len(str)<6 or len(str)>16:

return False

flag1=0

flag2=0

flag3=0

flag4=0

flag5=1

special\_char = ['$','@','#']

for char in str:

num = ord(char)

if num>=65 and num<=90:

flag1=1

elif num>=97 and num<=122:

flag2=1

elif num>=48 and num<=57:

flag3=1

elif char in special\_char:

flag4=1

else:

flag5=0

if flag1 and flag2 and flag3 and flag4 and flag5:

return True

return False

print("Enter password : ", end = " ")

password = input()

print(password)

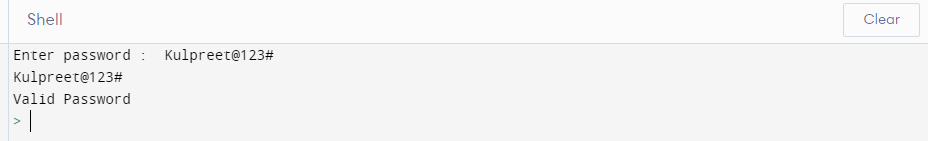
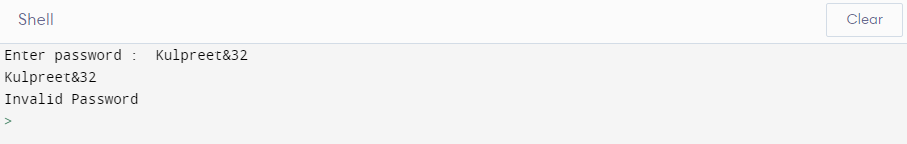
if isValid(password) :

print("Valid Password")

else:

print("Invalid Password")’

**OUTPUT:**

1. **Create a List L that is defined as= [10, 20, 30, 40, 50, 60, 70, 80].**
2. **WAP to add 200 and 300 to L.**
3. **WAP to remove 10 and 30 from L.**
4. **WAP to sort L in ascending order.**
5. **WAP to sort L in descending order.**

**CODE:**

l = [10, 20, 30, 40, 50, 60, 70, 80]

print(l)

l.append(200)

l.append(300)

print(l)

l.remove(10)

l.remove(30)

print(l)

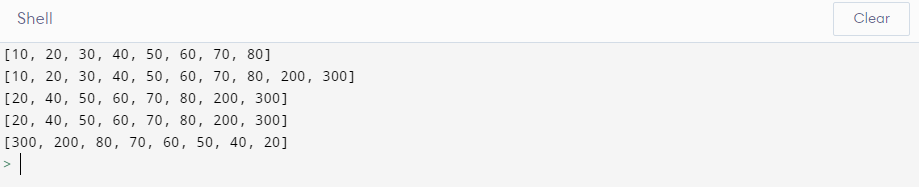
l.sort()

print(l)

l.sort(reverse=True)

print(l)

**OUTPUT:**



1. **D is a dictionary defined as D= {1:”One”, 2:”Two”, 3:”Three”, 4: “Four”, 5:”Five”}.**
2. **WAP to add new entry in D; key=6 and value is “Six”**
3. **WAP to remove key=2.**
4. **WAP to check if 6 key is present in D.**
5. **WAP to count the number of elements present in D.**
6. **WAP to add all the values present D.**

**CODE:**

D= {1:'One', 2:'Two', 3:'Three', 4: 'Four', 5:'Five'}

print('Original dictionary')

print(D)

D[6]='Six'

print("New Key added")

print(D)

D.pop(2)

print('Key deleted')

print(D)

if 6 in D:

print('Key 6 is present')

else:

print('Key is absent')

print('Total keys in dictionary = {0}'.format(len(D.keys())))

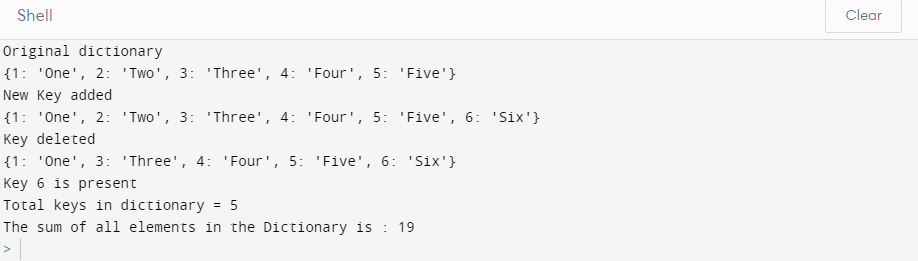
result=0

for key in D.keys():

result+=key

print(f'The sum of all elements in the Dictionary is : {result}')

**OUTPUT:**



1. **WAP to create a list of 100 random numbers between 100 and 900. Count and print the:**
2. **All odd numbers**
3. **All even numbers**
4. **All prime numbers**

**CODE:**

import random

import math

def isPrime(val):

flag = 0

for i in range(3,math.ceil(math.sqrt(val)),2):

if val%i==0:

flag = 1

break

if flag==0:

return True

return False

l = [random.randint(100,900) for i in range(100)]

odd=[]

even=[]

for num in l:

if num%2==0:

even.append(num)

else:

odd.append(num)

primes = [val for val in odd if isPrime(val)]

print('All odd numbers : ',odd)

print('Count of Odd numbers = ',len(odd))

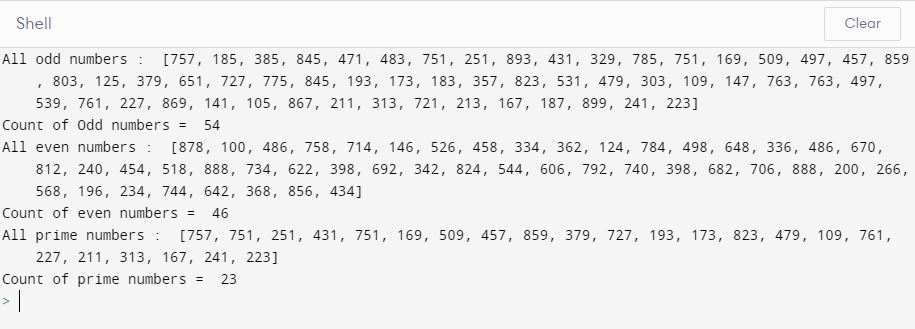
print('All even numbers : ',even)

print('Count of even numbers = ',len(even))

print('All prime numbers : ',primes)

print('Count of prime numbers = ',len(primes))

**OUTPUT:**



1. **Write a function which takes principal amount, interest rate and time. This function returns compound interest. Call this function to print the output**
2. **Save this function (as a module) in a python file and call it in another python file.**

**CODE:**

*#file name: interest.py*

def ciCalc(p, r, t, n):

return p \* pow((1 + r/(100 \* n)), n\*t)

*#file name: CompoundInterest.py*

import interest

p = float(input("\nEnter principal amount : "))

r = float(input("\nEnter annual rate of interest(in %) : "))

t = float(input("\nEnter time(in years) : "))

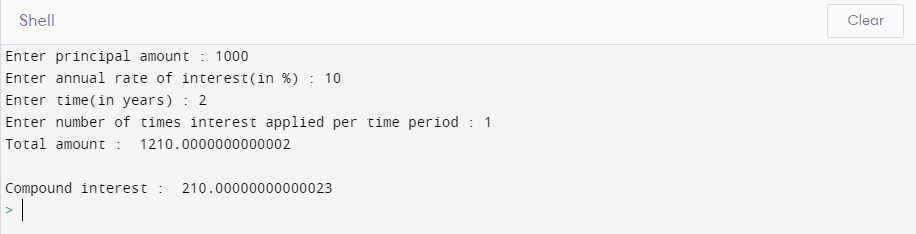
n = float(input("\nEnter number of times interest applied per time period : "))

a = interest.ciCalc(p,r,t,n)

print("\nTotal amount : ",a)

print("\nCompound interest : ",(a-p))

**OUTPUT:**



1. **Make a class called Restaurant. The \_\_init\_\_() method for Restaurant should store two attributes: a restaurant\_name and a cuisine\_type. Make a method called describe\_restaurant() that prints these two pieces of information, and a method called open\_restaurant() that prints a message indicating that the restaurant is open. Make an instance called restaurant from your class. Print the two attributes individually, and then call both methods.**

**CODE:**

class Restaurant:

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

self.restaurant\_name = name

self.cuisine\_type = ctype

def describe\_restaurant(self):

print("Restaurant Name :",self.restaurant\_name)

print("Cuisine type :",self.cuisine\_type)

def open\_restaurant(self):

print("The restaurant is OPEN")

restaurant = Restaurant("Elevate","Italian")

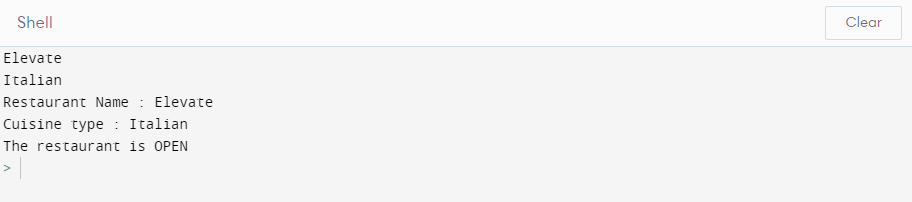
print(restaurant.restaurant\_name)

print(restaurant.cuisine\_type)

restaurant.describe\_restaurant()

restaurant.open\_restaurant()

**OUTPUT:**



1. **Make a class called User. Create two attributes called first\_name and last\_name, and then create several other attributes that are typically stored in a user profile. Make a method called describe\_user() that prints a summary of the user’s information. Make another method called greet\_user() that prints a personalized greeting to the user. Create several instances representing different users, and call both method for each user.**

**CODE:**

class User:

def \_\_init\_\_(self,fname,lname,roll,dob,college,degree):

self.first\_name = fname

self.last\_name = lname

self.roll\_number = roll

self.dob = dob

self.college\_name = college

self.degree = degree

def describe\_user(self):

print("\nUser Profile")

print("First Name :",self.first\_name)

print("Last Name :",self.last\_name)

print("Roll Number :",self.roll\_number)

print("Date of Birth :",self.dob)

print("College/University :",self.college\_name)

print("Degree/Programme :",self.degree)

def greet\_user(self):

print(f'\nHello {self.first\_name}! Hope you find Python interesting. Enjoy Learning!!')

user1 = User('Kulpreet','Singh',101803186,'01 September 2000','Thapar Institute of Engineering and Technology','B.Tech(COE)')

user1.describe\_user()

user1.greet\_user()

user2 = User('Ayush','Jain',101803690,'15 December 2000','Thapar Institute of Engineering and Technology','B.Tech(COE)')

user2.describe\_user()

user2.greet\_user()

**OUTPUT:**

