Lab Assignment-1

MANPREET SINGH , 102303357 , 2C25

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.

Results = {

'Maths': [87, 92, 78, 85, 90, 88, 76, 81, 95, 89],

'Science': [80, 85, 82, 88, 91, 84, 79, 83, 87, 86],

'English': [78, 82, 85, 88, 91, 84, 81, 79, 80, 86],

'IT': [92, 88, 91, 87, 85, 89, 86, 83, 84, 90]

}

test\_stats = {}

for subject, marks in results.items():

test\_stats[subject] = {

'highest': max(marks),

'lowest': min(marks),

'average': sum(marks) / len(marks)

}

all\_marks = [mark for marks in results.values() for mark in marks]

overall\_stats = {

'highest': max(all\_marks),

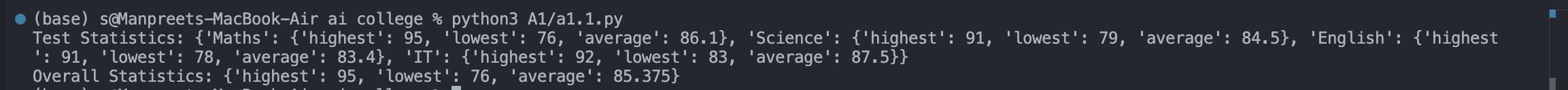
'lowest': min(all\_marks),

'average': sum(all\_marks) / len(all\_marks)

}

print("Test Statistics:", test\_stats)

print("Overall Statistics:", overall\_stats)



2)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%.

basic\_salary = float(input("Enter the basic salary: "))

if basic\_salary <= 10000:

hra = basic\_salary \* 0.20

da = basic\_salary \* 0.80

elif basic\_salary <= 20000:

hra = basic\_salary \* 0.25

da = basic\_salary \* 0.90

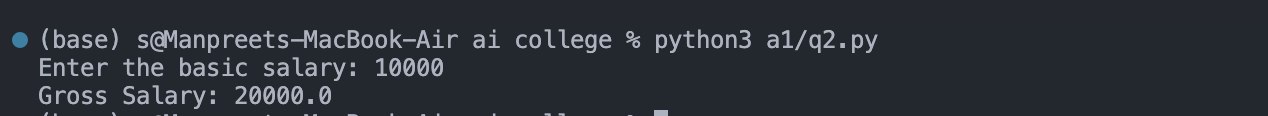
else:

hra = basic\_salary \* 0.30

da = basic\_salary \* 0.95

gross\_salary = basic\_salary + hra + da

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



3)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.

password = input("Enter the password: ")

is\_length\_valid = 6 <= len(password) <= 16

has\_lower = False

has\_upper = False

has\_digit = False

has\_special = False

for char in password:

if 'a' <= char <= 'z':

has\_lower = True

elif 'A' <= char <= 'Z':

has\_upper = True

elif '0' <= char <= '9':

has\_digit = True

elif char in "$#@":

has\_special = True

if not is\_length\_valid:

print("Password length must be between 6 and 16 characters.")

if not has\_lower:

print("Password must contain at least one lowercase letter.")

if not has\_upper:

print("Password must contain at least one uppercase letter.")

if not has\_digit:

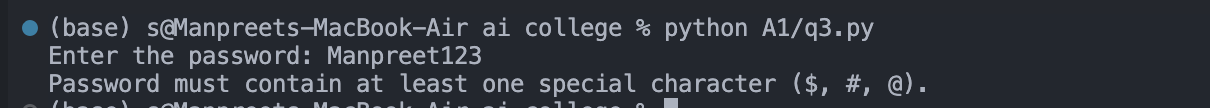
print("Password must contain at least one digit.")

if not has\_special:

print("Password must contain at least one special character ($, #, @).")

if is\_length\_valid and has\_lower and has\_upper and has\_digit and has\_special:

print("Password is valid.")



4)Create a List L that is defined as= [10, 20, 30, 40, 50, 60, 70, 80].

(i) WAP to add 200 and 300 to L.

(ii) WAP to remove 10 and 30 from L.

(iii) WAP to sort L in ascending order.

(iv) WAP to sort L in descending order.

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

L.append(200)

L.append(300)

L.remove(10)

L.remove(30)

L\_ascending = sorted(L)

L\_descending = sorted(L, reverse=True)

print(L)

print(L\_ascending)

print(L\_descending)



5) D is a dictionary defined as D= {1:”One”, 2:”Two”, 3:”Three”, 4: “Four”, 5:”Five”}.

(i) WAP to add new entry in D; key=6 and value is “Six”

(ii) WAP to remove key=2.

(iii) WAP to check if 6 key is present in D.

(iv) WAP to count the number of elements present in D.

(v) WAP to add all the values present D.

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

D[6] = "Six"

D.pop(2)

key\_present = 6 in D

element\_count = len(D)

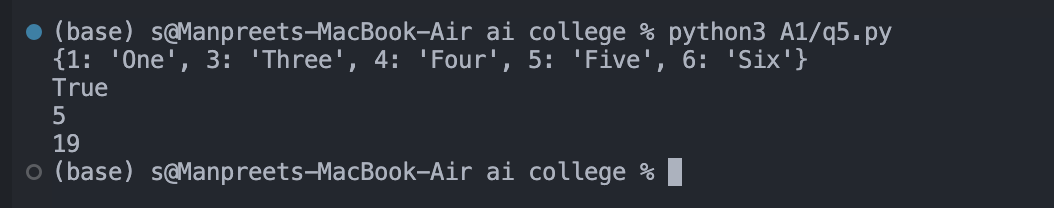
values\_combined = sum(D)

print(D)

print(key\_present)

print(element\_count)

print(values\_combined)



6) WAP to create a list of 100 random numbers between 100 and 900. Count and print the:

(i) All odd numbers

(ii) All even numbers

(iii) All prime numbers

import random

def is\_prime(n):

if n <= 1:

return False

for i in range(2, int(n \*\* 0.5) + 1):

if n % i == 0:

return False

return True

numbers = [random.randint(100, 900) for \_ in range(100)]

odd\_numbers = [num for num in numbers if num % 2 != 0]

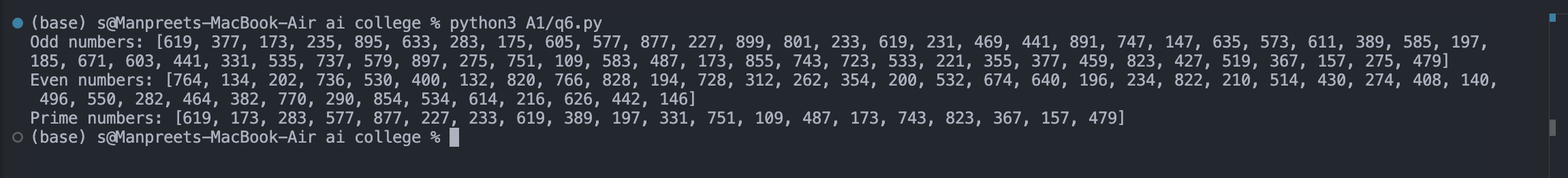
even\_numbers = [num for num in numbers if num % 2 == 0]

prime\_numbers = [num for num in numbers if is\_prime(num)]

print("Odd numbers:", odd\_numbers)

print("Even numbers:", even\_numbers)

print("Prime numbers:", prime\_numbers)



7) Write a function which takes principal amount, interest rate and time. This function returns

compound interest. Call this function to print the output.

(ii) Save this function (as a module) in a python file and call it in another python file.

from comp\_int import calculate\_compound\_interest

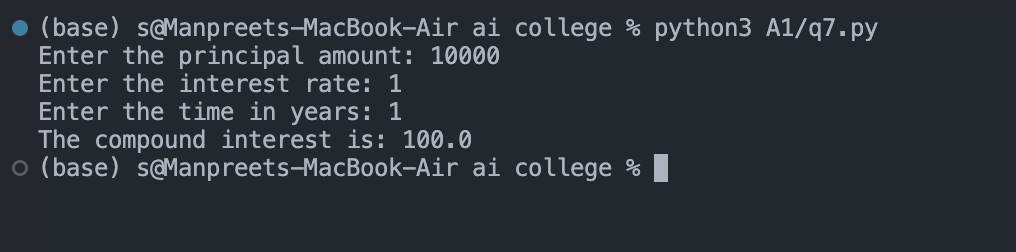
principal = float(input("Enter the principal amount: "))

rate = float(input("Enter the interest rate: "))

time = float(input("Enter the time in years: "))

interest = calculate\_compound\_interest(principal, rate, time)

print(f"The compound interest is: {interest}")



8) A) 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. B) 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.

class Restaurant:

def \_\_init\_\_(self, restaurant\_name, cuisine\_type):

self.restaurant\_name = restaurant\_name

self.cuisine\_type = cuisine\_type

def describe\_restaurant(self):

print(f"Restaurant Name: {self.restaurant\_name}")

print(f"Cuisine Type: {self.cuisine\_type}")

def open\_restaurant(self):

print(f"{self.restaurant\_name} is now open!")

restaurant = Restaurant("abc", "italian")

print(restaurant.restaurant\_name)

print(restaurant.cuisine\_type)

restaurant.describe\_restaurant()

restaurant.open\_restaurant()

class User:

def \_\_init\_\_(self, first\_name, last\_name, age, location):

self.first\_name = first\_name

self.last\_name = last\_name

self.age = age

self.location = location

def describe\_user(self):

print(f"User: {self.first\_name} {self.last\_name}")

print(f"Age: {self.age}")

print(f"Location: {self.location}")

def greet\_user(self):

print(f"Hello, {self.first\_name} {self.last\_name}! Welcome!")

user1 = User("Manp", "Singh", 20, "jalandhar")

user2 = User("xyz", "abc", 31, "delhi")

user1.describe\_user()

user1.greet\_user()

user2.describe\_user()

user2.greet\_user()

