FDS Assignments

1. Reverse an array.

def reverse(r\_list: list) -> list:

    return [r\_list[len(r\_list)-i] for i in range(1,len(r\_list)+1)]

a=[1,2,3,4,5]

a= reverse(a)

print(a)

2.Student data management

def take\_input(k):

     name  = input(f"Enter the name of the student {k+1}:")

     roll\_no =  input(f"Input the roll number of  {name} :")

     marks\_list = [int(input(f"Enter the marks for subject {i+1} :"))for i in range(5)]

     student = Student(name,roll\_no,marks\_list)

     print(f"The percentage of {student.name} is {student.percentage()}")

class Student:

     all\_student\_info=[]

     def \_\_init\_\_(self,name,roll\_no,list\_of\_marks):

         self.name = name

         self.roll\_no = roll\_no

         self.list\_of\_marks  = list\_of\_marks

         Student.all\_student\_info.append((name,roll\_no,sum(list\_of\_marks)))

     def percentage(self):

         return (sum(self.list\_of\_marks)/500)\*100

     def display\_toppers():

         Student.all\_student\_info.sort(key=lambda x:x[2])

         print(f"The rank one holder is {Student.all\_student\_info[-1]}\nRank two holder is {Student.all\_student\_info[-2]}")

n=int(input("Enter the number of students :"))

for i in range(n):

     take\_input(i)

Student.display\_toppers()

3.Account Management

class Account:

    all\_accounts=[]

    def \_\_init\_\_(self,name,account\_number,account\_balance:int):

        self.name=name

        self.account\_balance = account\_balance

        self.account\_number = account\_number

    def Credit(self,amount):

        self.account\_balance = amount +self.account\_balance

    def Debit(self,amount):

        if amount>self.account\_balance:

            print("Debit amount exceeded balance amount")

        else :

            self.account\_balance = self.account\_balance-amount

    def getBalance(self):

        print(f"Current Balance : {self.account\_balance}\nName of account holder : {self.name}\nAccount Number:{self.account\_number}")

n = int(input("The number of accounts : "))

for i in range(n):

    name = input("Enter your name :")

    account\_number= input("Enter the account number :")

    account\_balance = int(input("Enter the current account balance :"))

    account= Account(name,account\_number,account\_balance)

    Account.all\_accounts.append((account,account\_number))

for i in range(n):

    Account.all\_accounts[i][0].getBalance()

4.Pseudocode for finding GCD of 2 numbers.

Algorithm GCD (A, B) {

// A and B are two numbers

1.if (A>B) then

1.1 small = B;

else

1.1 small = A;

2.for i:= 0 to small+1 do

2.1 if ((A%i == 0) and (B%i == 0)) then

2.1.1 gcd = i;

3.write gcd;

}

5. Flowchart for the student problem

