**WEEK-1**

**Data structures and Algorithms**

**1.** **E-commerce Platform Search Function**

import java.util.\*;

class Product{

    int id;

    String name,category;

    double price;

    Product(int id,String name,String category,double price){

        this.id=id;

        this.name=name.toLowerCase();

        this.category=category.toLowerCase();

        this.price=price;

    }

boolean matches(String keyword){

        keyword=keyword.toLowerCase();

        return name.contains(keyword) || category.contains(keyword);

    }

    public String toString(){

        return id+":"+name+"("+category+")-"+price;

    }}

public class EcommerceSearch{

    public static void main(String[] args) {

        List<Product> products = Arrays.asList(

            new Product(1, "iPhone 14","Electronics", 80000),

            new Product(3, "Sony Headphones","Audio", 25000),

            new Product(4, "Nike Shoes","Footwear", 5000)

        );

        Scanner sc=new Scanner(System.in);

        System.out.print("Search:");

        String keyword=sc.nextLine();

        boolean found=false;

        for(Product p:products){

            if(p.matches(keyword)){

                System.out.println(p);

                found=true;

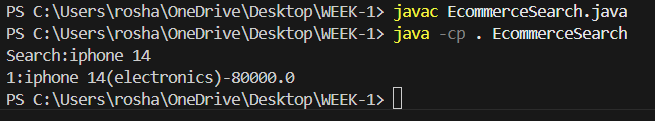
            }}

        if(!found)System.out.println("No products found");

        sc.close();

    }}

**Output**

****

**2. Financial Forecasting**

import java.util.\*;

public class FinancialForecast {

static double sum(List<Double> data,int k,int i){

if (k==0 || i<0) return 0;

return data.get(i) + sum(data,k-1,i-1);

}

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

List<Double> rev=new ArrayList<>();

for (int i=0;i<n;i++) rev.add(sc.nextDouble());

int k=sc.nextInt();

if (k>n)

System.out.println("Insufficient data");

else{

double forecast=sum(rev,k,n-1)/k;

System.out.printf("%.2f\n", forecast);

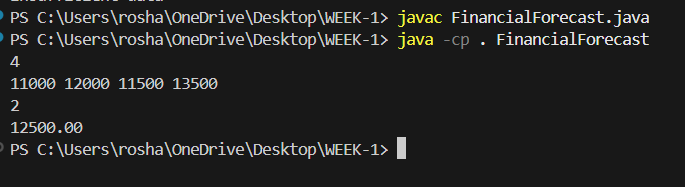
}

sc.close();

}

}

**Output**

****