**Week1\_DataStructuresAndAlgorithms\_HandsOn**

**Exercise 2: E-commerce Platform Search Function :**

**Code:**

import java.util.\*;

class Product{

int productId;

String productName;

String category;

Product(int id,String name,String category)

{

this.productId=id;

this.productName=name;

this.category=category;

}

}

class Ecommerce{

public static int linearSearch(Product[] product,String target)

{

for(int i=0;i<product.length;i++)

{

if(product[i].productName.equalsIgnoreCase(target))

{

return i;

}

}

return -1;

}

public static int BinarySearch(Product[] product,String target)

{

int l=0,r=product.length-1;

while(l<=r)

{

int m=(l+r)/2;

int cmp=product[m].productName.compareToIgnoreCase(target);

if(cmp==0){

return m;

}

else if(cmp<0)

{

l=m+1;

}

else{

r=m-1;

}

}

return -1;

}

}

class Main{

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

int n=s.nextInt();

Product[] product=new Product[n];

for(int i=0;i<n;i++)

{

System.out.println("Enter Product ID : ");

int id=s.nextInt();

s.nextLine();

System.out.println("Enter Product Name : ");

String name=s.nextLine();

System.out.println("Enter Product Category : ");

String category=s.nextLine();

product[i]=new Product(id,name,category);

}

System.out.println("Enter Product to be Searched :");

String target=s.nextLine();

int linearIndex=Ecommerce.linearSearch(product,target);

if (linearIndex != -1) {

System.out.println("Linear Search: Product found at index " + linearIndex);

} else {

System.out.println("Linear Search: Product not found");

}

Arrays.sort(product,Comparator.comparing(p->p.productName.toLowerCase()));

int BinaryIndex=Ecommerce.BinarySearch(product,target);

if (BinaryIndex != -1) {

System.out.println("Binary Search: Product found at index " + BinaryIndex);

} else {

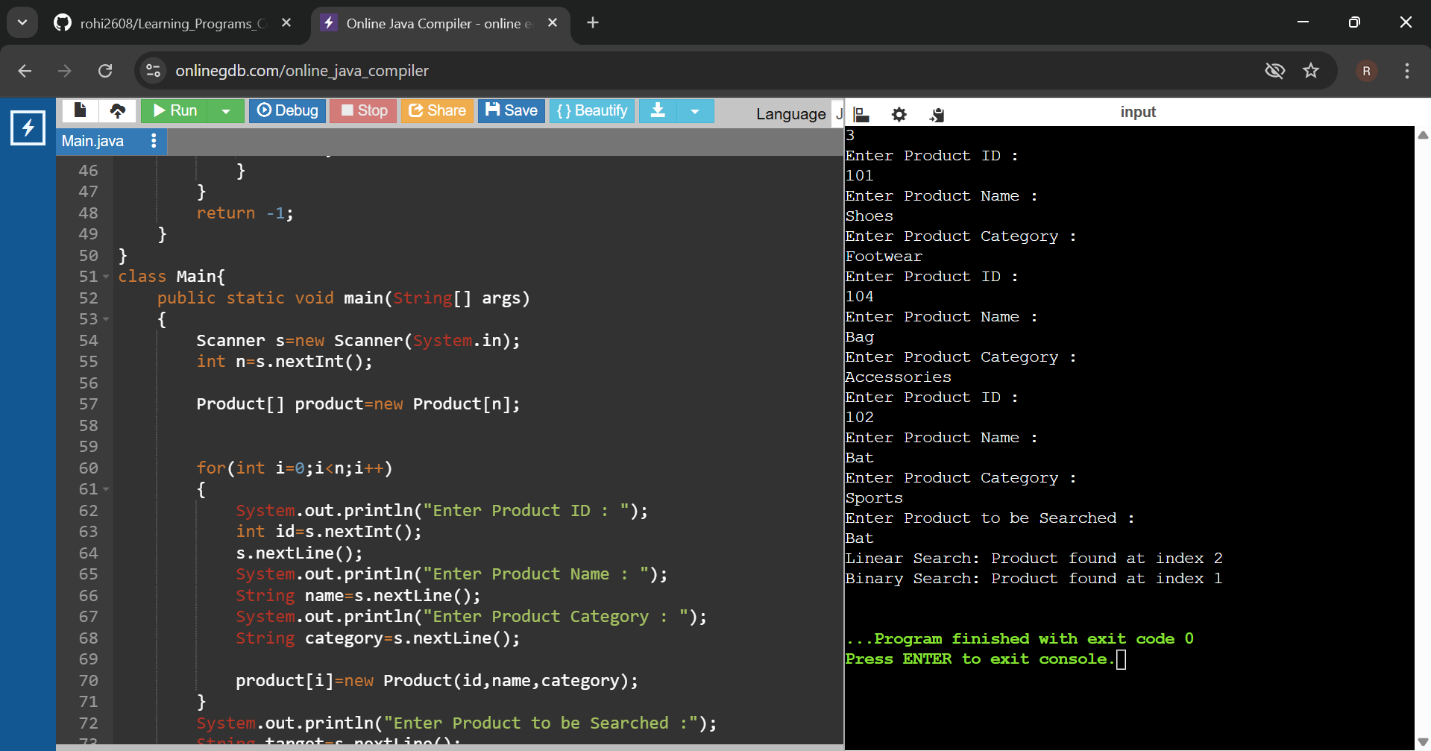
System.out.println("Binary Search: Product not found");

}

}

}

**Output:**



**Exercise 7: Financial Forecasting :**

**Code:**

import java.util.\*;

class Main{

public static double futurevalue(double presentvalue,double rate,int years)

{

if(years==0)

{

return presentvalue;

}

return (1+rate)\* futurevalue(presentvalue,rate,years-1);

}

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

System.out.print("Enter present value: ");

double presentvalue = s.nextDouble();

System.out.print("Enter annual growth rate : ");

double rate = s.nextDouble();

System.out.print("Enter number of years: ");

int years = s.nextInt();

double predictedvalue = futurevalue(presentvalue, rate, years);

System.out.print("Predicted future value is : "+predictedvalue);

}

}

**Output:**

