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

**Scenario:**

You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.

**THEORY:**

Implemented searches such that the code says found only when the product attribute is found through both the linear and binary search.

**CODE:**

import java.util.Arrays;

import java.util.Comparator;

import java.util.Scanner;

public class Ecommerce {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        int n = scan.nextInt();

        Product[] a = new Product[n];

        for(int i=0;i<n;i++){

            int id = scan.nextInt();

            String name = scan.next();

            String cat = scan.next();

            a[i] = new Product(id, name, cat);

        }

        System.out.println("Enter the attribute to be searched");

        String search = scan.next();

        if(search.equals("id")){

            int num = scan.nextInt();

            Search(a,num,null,null);

        }

        else if(search.equals("name")){

            String attr1 = scan.next();

            Search(a,0,attr1,null);

        }

        else{

            String attr2 = scan.next();

            Search(a,0,null,attr2);

        }

        scan.close();

    }

    static void Search(Product[] ar,int n,String name,String cat){

        if(linearSearch(ar,n,name,cat) && binarySearch(ar,n,name,cat)){

            System.out.println("Product found");

        }

        else{

            System.out.println("Product not found");

        }

    }

    static boolean linearSearch(Product[] a,int n,String name,String cat){

        if(n == 0){

            if(name == null){

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

                    if(a[i].category.equals(cat)){

                        return true;

                    }

                }

            }

            else{

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

                    if(a[i].productName.equals(name)){

                        return true;

                    }

                }

            }

        }

        else{

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

                if(a[i].productId == n){

                    return true;

                }

            }

        }

        return false;

    }

static boolean binarySearch(Product[] a, int id, String name, String cat) {

        if (id != 0) {

            Arrays.sort(a, Comparator.comparingInt(p -> p.productId));

            int low = 0, high = a.length - 1;

            while (low <= high) {

                int mid = (low + high) / 2;

                if (a[mid].productId == id) return true;

                else if (a[mid].productId < id) low = mid + 1;

                else high = mid - 1;

            }

        }

        else if (name != null) {

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

            int low = 0, high = a.length - 1;

            while (low <= high) {

                int mid = (low + high) / 2;

                int cmp = a[mid].productName.compareTo(name);

                if (cmp == 0) return true;

                else if (cmp < 0) low = mid + 1;

                else high = mid - 1;

            }

        }

        else if (cat != null) {

            Arrays.sort(a, Comparator.comparing(p -> p.category));

            int low = 0, high = a.length - 1;

            while (low <= high) {

                int mid = (low + high) / 2;

                int cmp = a[mid].category.compareTo(cat);

                if (cmp == 0) return true;

                else if (cmp < 0) low = mid + 1;

                else high = mid - 1;

            }

        }

        return false;

    }

}

class Product{

    int productId;

    String productName;

    String category;

    public Product(int id,String name,String cat){

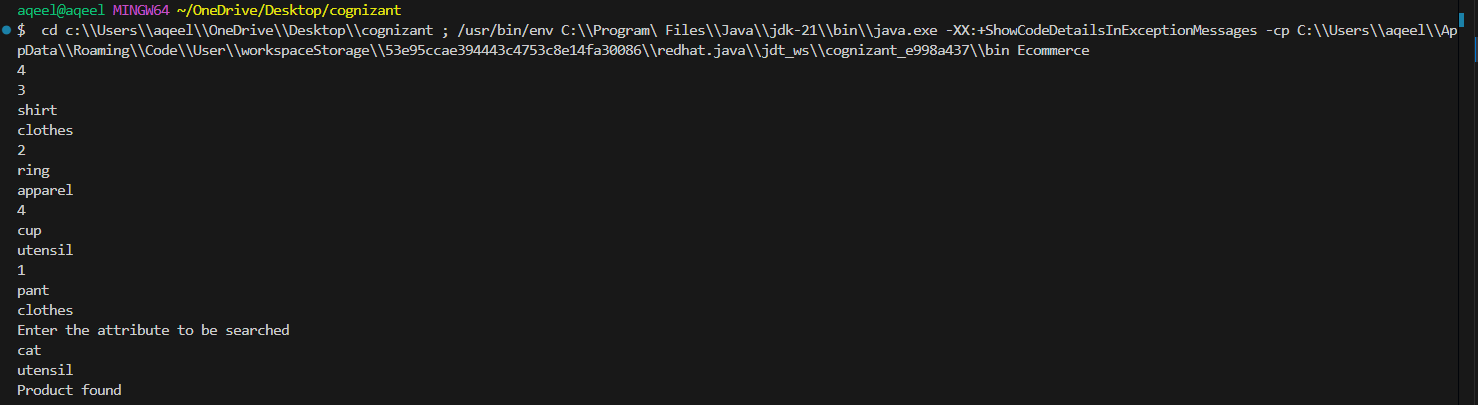
        this.productId = id;

        this.productName = name;

        this.category = cat;

    } }

**OUTPUT:**

****