using System;

public class Product {

    public int ProdId { get; set; }

    public string ProdName { get; set; }

    public string Category { get; set; }

    public Product(int id,string name,string category) {

        ProdId=id;

        ProdName=name;

        Category=category;

    }

    public override string ToString() {

        return $"[{ProdId}] {ProdName} - {Category}";

    }

}

class Program {

    static void Main(string[] args) {

        Product[] products=new Product[] {

            new Product(104,"Apple","Electronics"),

            new Product(101,"Nike","Fashion"),

            new Product(105,"Samsung","Electronics"),

            new Product(102,"Adidas","Fashion"),

            new Product(103,"Headphones","Accessories")

        };

        Console.Write("Enter product ID: ");

        int targetId=int.Parse(Console.ReadLine());

        Console.WriteLine("\nLinear Search Algorithm O(N): ");

        Product result1=LinearSearch(products,targetId);

        if (result1 != null) {

             Console.WriteLine(result1.ToString());

        }

        else{

            Console.WriteLine("Not found");

        }

        Array.Sort(products,(a,b)=>a.ProdId.CompareTo(b.ProdId));

        Console.WriteLine("\nBinary Search Algorithm O(log N): ");

        Product result2=BinarySearch(products,targetId);

        if (result2 != null) {

             Console.WriteLine(result2.ToString());

        }

        else {

            Console.WriteLine("Not found");

        }

    }

    static Product LinearSearch(Product[] products,int targetId) {

        foreach (var product in products) {

            if (product.ProdId==targetId) return product;

        }

        return null;

    }

    static Product BinarySearch(Product[] products,int targetId) {

        int left=0,right=products.Length-1;

        while (left<=right) {

            int mid=(left+right)/2;

            if (products[mid].ProdId==targetId) return products[mid];

            else if (products[mid].ProdId<targetId) left=mid+1;

            else right=mid-1;

        }

        return null;

    }

}