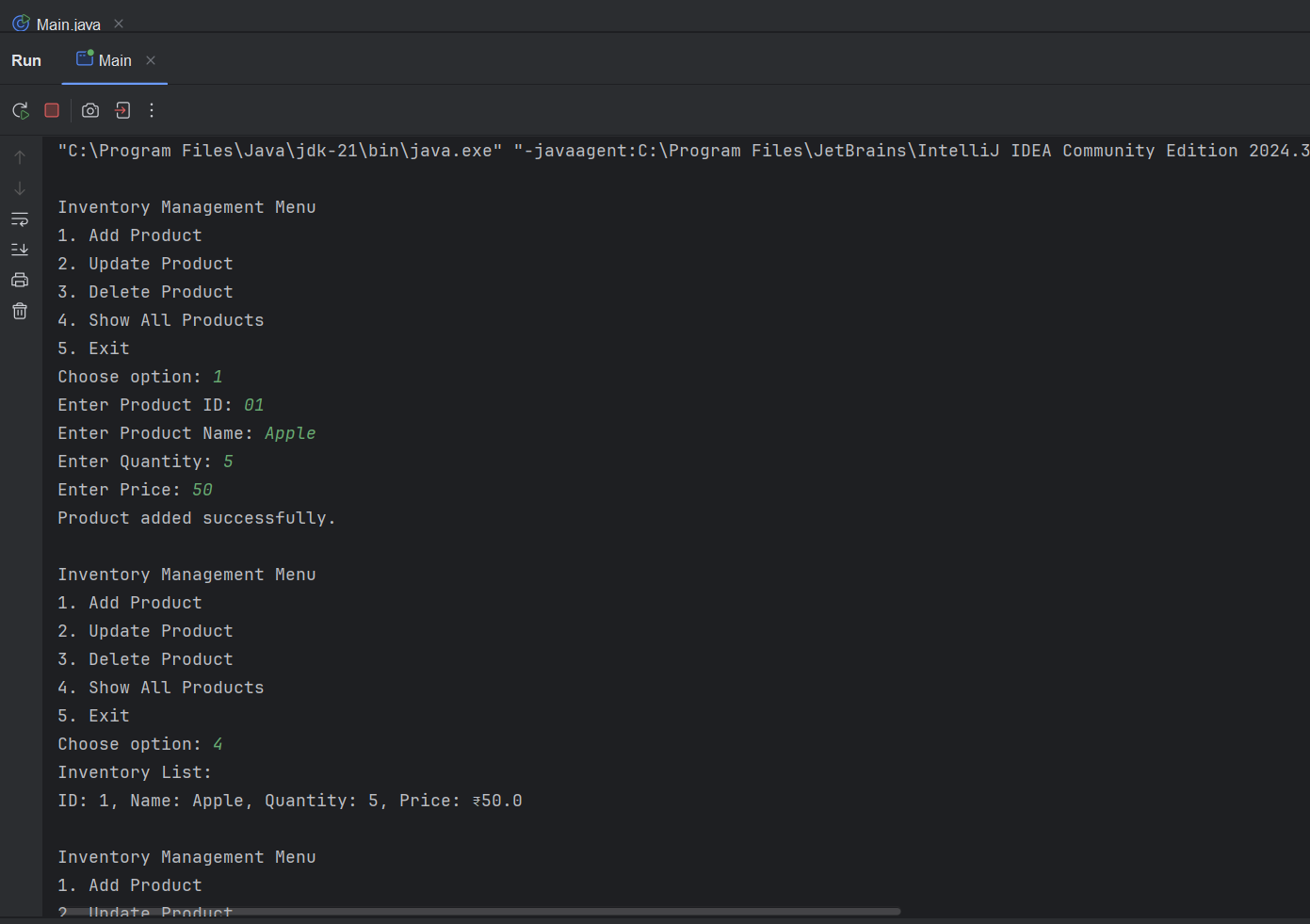
**Exercise 1: Inventory Management System**

**Code:**import java.util.\*;  
  
class Product {  
 int productId;  
 String productName;  
 int quantity;  
 double price;  
  
 public Product(int productId, String productName, int quantity, double price) {  
 this.productId = productId;  
 this.productName = productName;  
 this.quantity = quantity;  
 this.price = price;  
 }  
  
 public String toString() {  
 return "ID: " + productId + ", Name: " + productName +  
 ", Quantity: " + quantity + ", Price: ₹" + price;  
 }  
}  
  
public class Main {  
 static HashMap<Integer, Product> *inventory* = new HashMap<>();  
 static Scanner *sc* = new Scanner(System.*in*);  
  
 public static void main(String[] args) {  
 while (true) {  
 System.*out*.println("\nInventory Management Menu");  
 System.*out*.println("1. Add Product");  
 System.*out*.println("2. Update Product");  
 System.*out*.println("3. Delete Product");  
 System.*out*.println("4. Show All Products");  
 System.*out*.println("5. Exit");  
 System.*out*.print("Choose option: ");  
 int choice = *sc*.nextInt();  
  
 switch (choice) {  
 case 1: *addProduct*(); break;  
 case 2: *updateProduct*(); break;  
 case 3: *deleteProduct*(); break;  
 case 4: *showInventory*(); break;  
 case 5: System.*exit*(0);  
 default: System.*out*.println("Invalid option.");  
 }  
 }  
 }  
  
 static void addProduct() {  
 System.*out*.print("Enter Product ID: ");  
 int id = *sc*.nextInt();  
 *sc*.nextLine();  
 System.*out*.print("Enter Product Name: ");  
 String name = *sc*.nextLine();  
 System.*out*.print("Enter Quantity: ");  
 int qty = *sc*.nextInt();  
 System.*out*.print("Enter Price: ");  
 double price = *sc*.nextDouble();  
  
 Product p = new Product(id, name, qty, price);  
 *inventory*.put(id, p);  
 System.*out*.println("Product added successfully.");  
 }  
  
 static void updateProduct() {  
 System.*out*.print("Enter Product ID to update: ");  
 int id = *sc*.nextInt();  
 if (*inventory*.containsKey(id)) {  
 Product p = *inventory*.get(id);  
 System.*out*.print("Enter new Quantity: ");  
 p.quantity = *sc*.nextInt();  
 System.*out*.print("Enter new Price: ");  
 p.price = *sc*.nextDouble();  
 System.*out*.println("Product updated.");  
 } else {  
 System.*out*.println("Product not found.");  
 }  
 }  
  
 static void deleteProduct() {  
 System.*out*.print("Enter Product ID to delete: ");  
 int id = *sc*.nextInt();  
 if (*inventory*.remove(id) != null) {  
 System.*out*.println("Product deleted.");  
 } else {  
 System.*out*.println("Product not found.");  
 }  
 }  
  
 static void showInventory() {  
 if (*inventory*.isEmpty()) {  
 System.*out*.println("Inventory is empty.");  
 } else {  
 System.*out*.println("Inventory List:");  
 for (Product p : *inventory*.values()) {  
 System.*out*.println(p);  
 }  
 }  
 }  
}

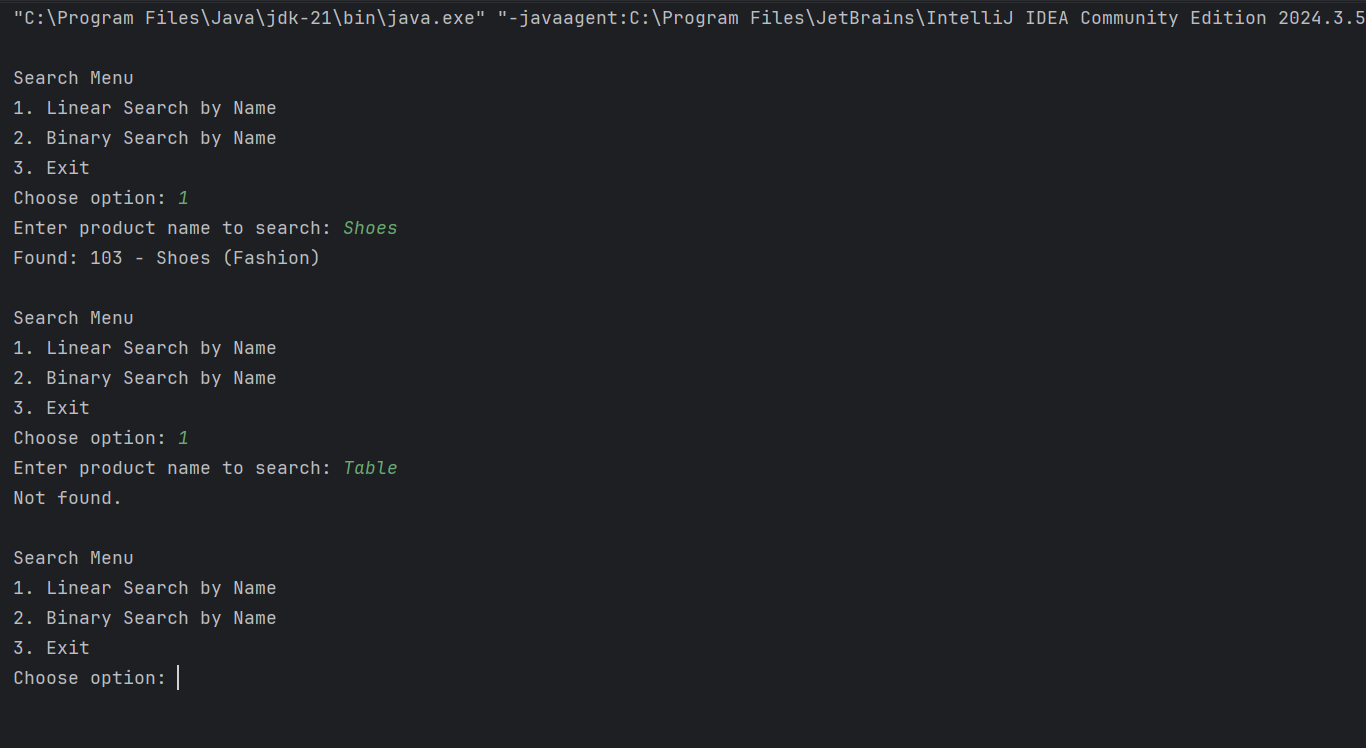
**Output:**



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

**Code:**

import java.util.\*;  
  
class Product {  
 int productId;  
 String productName;  
 String category;  
  
 Product(int productId, String productName, String category) {  
 this.productId = productId;  
 this.productName = productName;  
 this.category = category;  
 }  
  
 public String toString() {  
 return productId + " - " + productName + " (" + category + ")";  
 }  
}  
  
public class Main {  
 static Product[] *products*;  
  
 public static void main(String[] args) {  
 *products* = new Product[] {  
 new Product(101, "Laptop", "Electronics"),  
 new Product(102, "Keyboard", "Electronics"),  
 new Product(103, "Shoes", "Fashion"),  
 new Product(104, "Chair", "Furniture"),  
 new Product(105, "Lamp", "Home Decor")  
 };  
  
 Scanner s = new Scanner(System.*in*);  
  
 while(true) {  
 System.*out*.println("\nSearch Menu");  
 System.*out*.println("1. Linear Search by Name");  
 System.*out*.println("2. Binary Search by Name");  
 System.*out*.println("3. Exit");  
 System.*out*.print("Choose option: ");  
 int ch = s.nextInt();  
 s.nextLine();  
  
 if(ch == 1) {  
 System.*out*.print("Enter product name to search: ");  
 String name = s.nextLine();  
 boolean found = false;  
 for(Product p : *products*) {  
 if(p.productName.equalsIgnoreCase(name)) {  
 System.*out*.println("Found: " + p);  
 found = true;  
 break;  
 }  
 }  
 if(!found) System.*out*.println("Not found.");  
 }  
  
 else if(ch == 2) {  
 Arrays.*sort*(*products*, Comparator.*comparing*(p -> p.productName.toLowerCase()));  
 System.*out*.print("Enter product name to search: ");  
 String name = s.nextLine();  
 int low = 0, high = *products*.length - 1;  
 boolean found = false;  
  
 while(low <= high) {  
 int mid = (low + high) / 2;  
 int cmp = *products*[mid].productName.compareToIgnoreCase(name);  
 if(cmp == 0) {  
 System.*out*.println("Found: " + *products*[mid]);  
 found = true;  
 break;  
 } else if(cmp < 0) {  
 low = mid + 1;  
 } else {  
 high = mid - 1;  
 }  
 }  
  
 if(!found) System.*out*.println("Not found.");  
 }  
  
 else if(ch == 3) {  
 break;  
 }  
  
 else {  
 System.*out*.println("Invalid choice.");  
 }  
 }  
 }  
} **Output:**

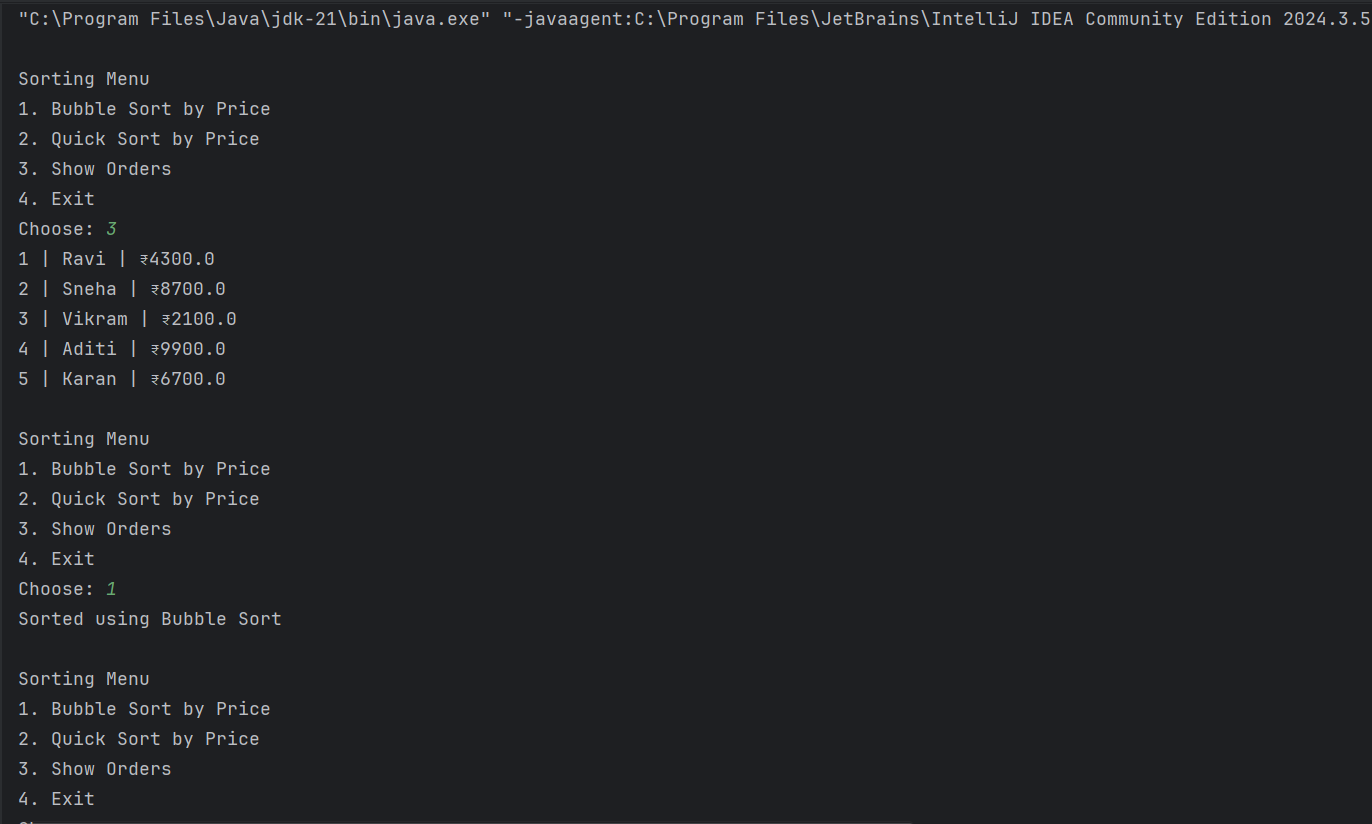
****

**Exercise 3: Sorting Customer Orders**

**Code:**

import java.util.\*;  
  
class Order {  
 int orderId;  
 String customerName;  
 double totalPrice;  
  
 Order(int orderId, String customerName, double totalPrice) {  
 this.orderId = orderId;  
 this.customerName = customerName;  
 this.totalPrice = totalPrice;  
 }  
  
 public String toString() {  
 return orderId + " | " + customerName + " | ₹" + totalPrice;  
 }  
}  
  
public class Main {  
 static Order[] *orders*;  
  
 public static void main(String[] args) {  
 *orders* = new Order[] {  
 new Order(1, "Ravi", 4300),  
 new Order(2, "Sneha", 8700),  
 new Order(3, "Vikram", 2100),  
 new Order(4, "Aditi", 9900),  
 new Order(5, "Karan", 6700)  
 };  
  
 Scanner sc = new Scanner(System.*in*);  
  
 while(true) {  
 System.*out*.println("\nSorting Menu");  
 System.*out*.println("1. Bubble Sort by Price");  
 System.*out*.println("2. Quick Sort by Price");  
 System.*out*.println("3. Show Orders");  
 System.*out*.println("4. Exit");  
 System.*out*.print("Choose: ");  
 int ch = sc.nextInt();  
  
 if(ch == 1) {  
 *bubbleSort*(*orders*);  
 System.*out*.println("Sorted using Bubble Sort");  
 }  
  
 else if(ch == 2) {  
 *quickSort*(*orders*, 0, *orders*.length - 1);  
 System.*out*.println("Sorted using Quick Sort");  
 }  
  
 else if(ch == 3) {  
 for(Order o : *orders*) System.*out*.println(o);  
 }  
  
 else if(ch == 4) break;  
  
 else System.*out*.println("Invalid");  
 }  
 }  
  
 static void bubbleSort(Order[] arr) {  
 for(int i = 0; i < arr.length-1; i++) {  
 for(int j = 0; j < arr.length-i-1; j++) {  
 if(arr[j].totalPrice > arr[j+1].totalPrice) {  
 Order temp = arr[j];  
 arr[j] = arr[j+1];  
 arr[j+1] = temp;  
 }  
 }  
 }  
 }  
  
 static void quickSort(Order[] arr, int low, int high) {  
 if(low < high) {  
 int pi = *partition*(arr, low, high);  
 *quickSort*(arr, low, pi-1);  
 *quickSort*(arr, pi+1, high);  
 }  
 }  
  
 static int partition(Order[] arr, int low, int high) {  
 double pivot = arr[high].totalPrice;  
 int i = low - 1;  
 for(int j = low; j < high; j++) {  
 if(arr[j].totalPrice < pivot) {  
 i++;  
 Order temp = arr[i];  
 arr[i] = arr[j];  
 arr[j] = temp;  
 }  
 }  
 Order temp = arr[i+1];  
 arr[i+1] = arr[high];  
 arr[high] = temp;  
 return i+1;  
 }  
}

**Output:**

****

**Exercise 4: Employee Management System**

**Code:**

import java.util.\*;  
  
class Employee {  
 int employeeId;  
 String name;  
 String position;  
 double salary;  
  
 Employee(int employeeId, String name, String position, double salary) {  
 this.employeeId = employeeId;  
 this.name = name;  
 this.position = position;  
 this.salary = salary;  
 }  
  
 public String toString() {  
 return employeeId + " | " + name + " | " + position + " | ₹" + salary;  
 }  
}  
  
public class Main {  
 static Employee[] *employees* = new Employee[100];  
 static int *count* = 0;  
 static Scanner *sc* = new Scanner(System.*in*);  
  
 public static void main(String[] args) {  
 while(true) {  
 System.*out*.println("\nEmployee Management Menu");  
 System.*out*.println("1. Add Employee");  
 System.*out*.println("2. Search Employee by ID");  
 System.*out*.println("3. Show All Employees");  
 System.*out*.println("4. Delete Employee by ID");  
 System.*out*.println("5. Exit");  
 System.*out*.print("Choose: ");  
 int ch = *sc*.nextInt();  
  
 if(ch == 1) {  
 System.*out*.print("ID: ");  
 int id = *sc*.nextInt();  
 *sc*.nextLine();  
 System.*out*.print("Name: ");  
 String name = *sc*.nextLine();  
 System.*out*.print("Position: ");  
 String pos = *sc*.nextLine();  
 System.*out*.print("Salary: ");  
 double sal = *sc*.nextDouble();  
 *employees*[*count*++] = new Employee(id, name, pos, sal);  
 System.*out*.println("Added");  
 }  
  
 else if(ch == 2) {  
 System.*out*.print("Enter ID to search: ");  
 int id = *sc*.nextInt();  
 boolean found = false;  
 for(int i = 0; i < *count*; i++) {  
 if(*employees*[i].employeeId == id) {  
 System.*out*.println("Found: " + *employees*[i]);  
 found = true;  
 break;  
 }  
 }  
 if(!found) System.*out*.println("Not found");  
 }  
  
 else if(ch == 3) {  
 if(*count* == 0) System.*out*.println("No records");  
 for(int i = 0; i < *count*; i++) System.*out*.println(*employees*[i]);  
 }  
  
 else if(ch == 4) {  
 System.*out*.print("Enter ID to delete: ");  
 int id = *sc*.nextInt();  
 boolean deleted = false;  
 for(int i = 0; i < *count*; i++) {  
 if(*employees*[i].employeeId == id) {  
 for(int j = i; j < *count* - 1; j++)  
 *employees*[j] = *employees*[j + 1];  
 *employees*[--*count*] = null;  
 deleted = true;  
 System.*out*.println("Deleted");  
 break;  
 }  
 }  
 if(!deleted) System.*out*.println("Not found");  
 }  
  
 else if(ch == 5) break;  
  
 else System.*out*.println("Invalid");  
 }  
 }  
}

**Output:**

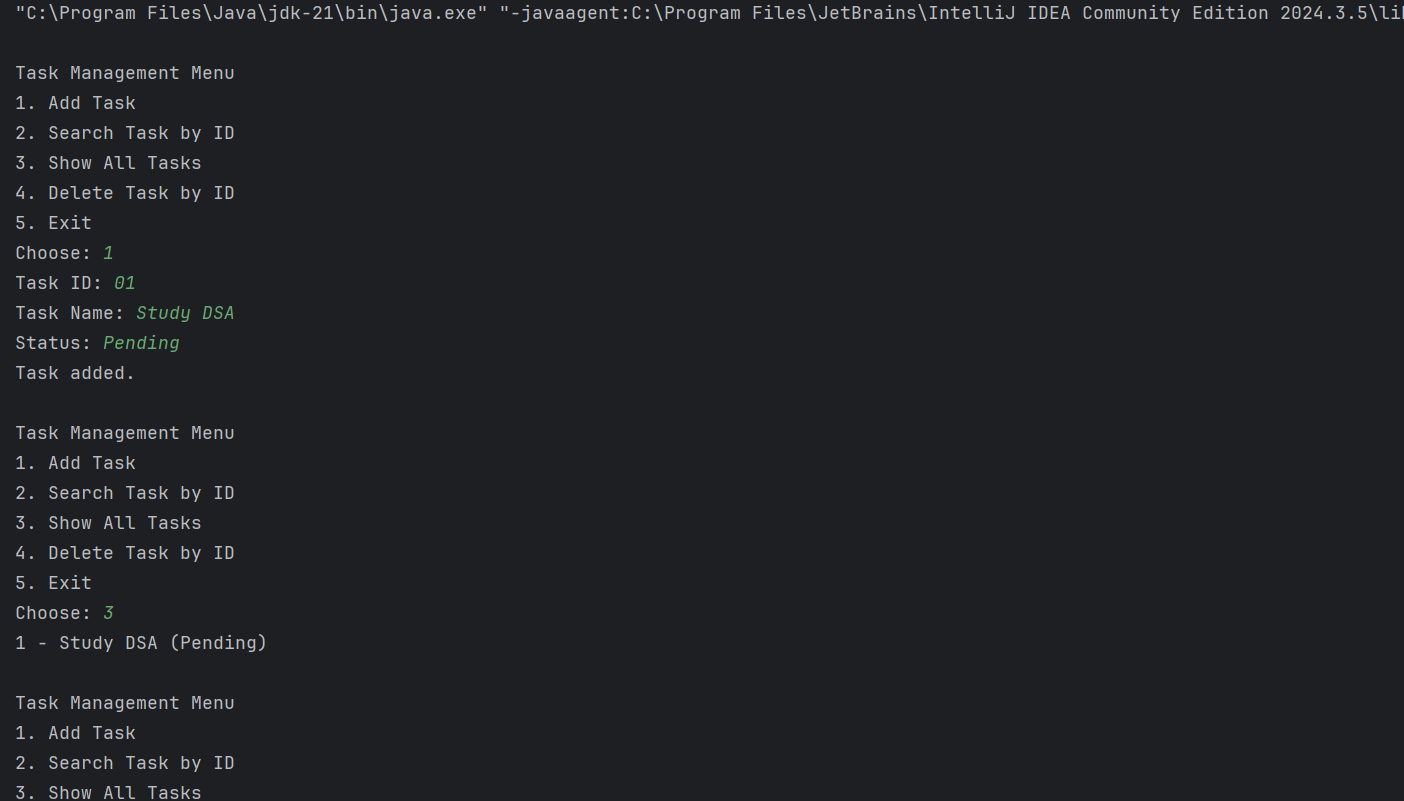
****

**Exercise 5: Task Management System (Linked List)**

**Code:**

import java.util.\*;  
  
class Task {  
 int taskId;  
 String taskName;  
 String status;  
 Task next;  
  
 Task(int taskId, String taskName, String status) {  
 this.taskId = taskId;  
 this.taskName = taskName;  
 this.status = status;  
 this.next = null;  
 }  
  
 public String toString() {  
 return taskId + " - " + taskName + " (" + status + ")";  
 }  
}  
  
public class Main {  
 static Task *head* = null;  
 static Scanner *sc* = new Scanner(System.*in*);  
  
 public static void main(String[] args) {  
 while(true) {  
 System.*out*.println("\nTask Management Menu");  
 System.*out*.println("1. Add Task");  
 System.*out*.println("2. Search Task by ID");  
 System.*out*.println("3. Show All Tasks");  
 System.*out*.println("4. Delete Task by ID");  
 System.*out*.println("5. Exit");  
 System.*out*.print("Choose: ");  
 int ch = *sc*.nextInt();  
  
 if(ch == 1) {  
 System.*out*.print("Task ID: ");  
 int id = *sc*.nextInt();  
 *sc*.nextLine();  
 System.*out*.print("Task Name: ");  
 String name = *sc*.nextLine();  
 System.*out*.print("Status: ");  
 String status = *sc*.nextLine();  
 Task newTask = new Task(id, name, status);  
 newTask.next = *head*;  
 *head* = newTask;  
 System.*out*.println("Task added.");  
 }  
  
 else if(ch == 2) {  
 System.*out*.print("Enter Task ID to search: ");  
 int id = *sc*.nextInt();  
 Task curr = *head*;  
 boolean found = false;  
 while(curr != null) {  
 if(curr.taskId == id) {  
 System.*out*.println("Found: " + curr);  
 found = true;  
 break;  
 }  
 curr = curr.next;  
 }  
 if(!found) System.*out*.println("Not found.");  
 }  
  
 else if(ch == 3) {  
 if(*head* == null) {  
 System.*out*.println("No tasks.");  
 continue;  
 }  
 Task temp = *head*;  
 while(temp != null) {  
 System.*out*.println(temp);  
 temp = temp.next;  
 }  
 }  
  
 else if(ch == 4) {  
 System.*out*.print("Enter Task ID to delete: ");  
 int id = *sc*.nextInt();  
 Task temp = *head*, prev = null;  
 while(temp != null && temp.taskId != id) {  
 prev = temp;  
 temp = temp.next;  
 }  
  
 if(temp == null) System.*out*.println("Task not found.");  
 else {  
 if(prev == null) *head* = temp.next;  
 else prev.next = temp.next;  
 System.*out*.println("Task deleted.");  
 }  
 }  
  
 else if(ch == 5) break;  
  
 else System.*out*.println("Invalid choice");  
 }  
 }  
}

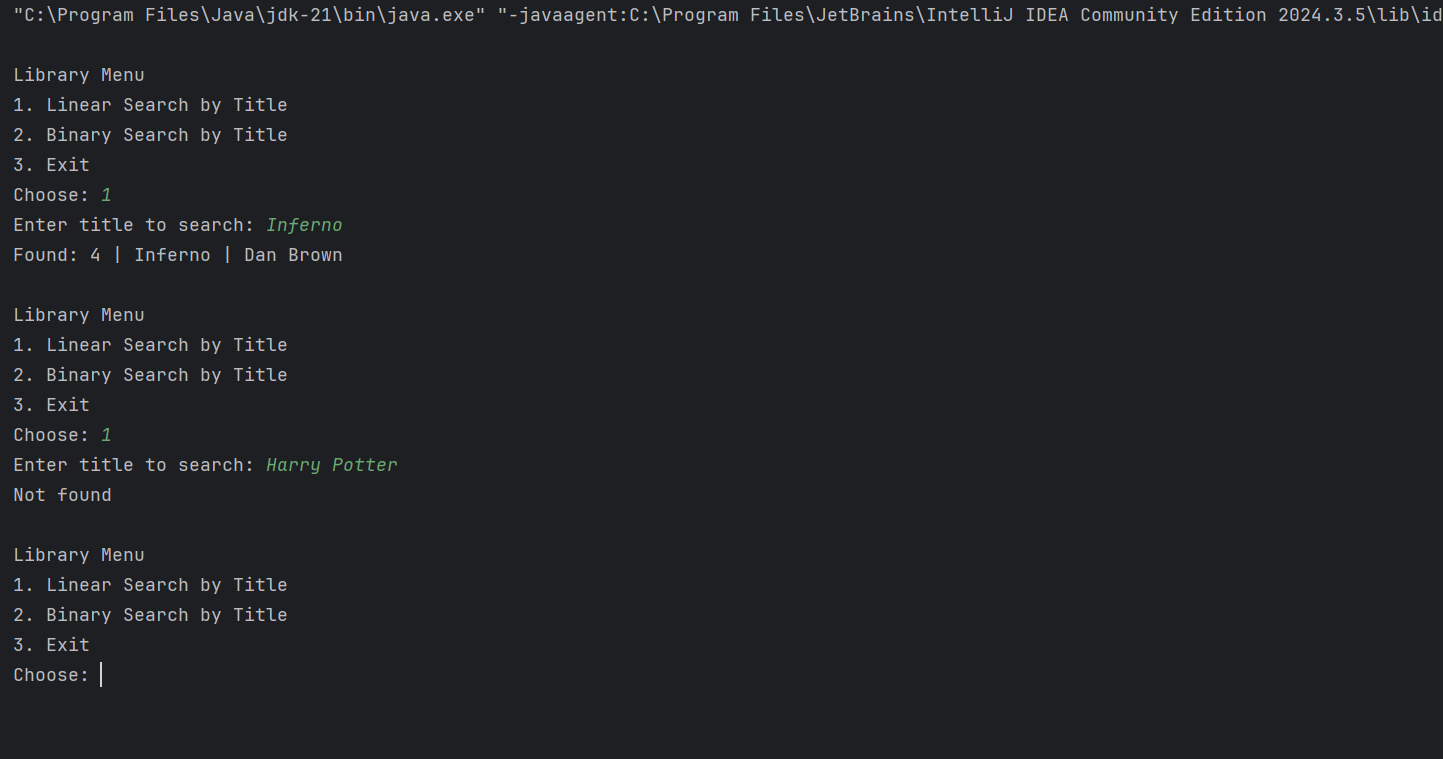
**Output:**

****

**Exercise 6: Library Management System (Search Books)**

**Code:**import java.util.\*;  
  
class Book {  
 int bookId;  
 String title;  
 String author;  
  
 Book(int bookId, String title, String author) {  
 this.bookId = bookId;  
 this.title = title;  
 this.author = author;  
 }  
  
 public String toString() {  
 return bookId + " | " + title + " | " + author;  
 }  
}  
  
public class Main {  
 static Book[] *books* = {  
 new Book(1, "The Alchemist", "Paulo Coelho"),  
 new Book(2, "Atomic Habits", "James Clear"),  
 new Book(3, "Clean Code", "Robert Martin"),  
 new Book(4, "Inferno", "Dan Brown"),  
 new Book(5, "1984", "George Orwell")  
 };  
  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
  
 while(true) {  
 System.*out*.println("\nLibrary Menu");  
 System.*out*.println("1. Linear Search by Title");  
 System.*out*.println("2. Binary Search by Title");  
 System.*out*.println("3. Exit");  
 System.*out*.print("Choose: ");  
 int ch = sc.nextInt();  
 sc.nextLine();  
  
 if(ch == 1) {  
 System.*out*.print("Enter title to search: ");  
 String key = sc.nextLine();  
 boolean found = false;  
 for(Book b : *books*) {  
 if(b.title.equalsIgnoreCase(key)) {  
 System.*out*.println("Found: " + b);  
 found = true;  
 break;  
 }  
 }  
 if(!found) System.*out*.println("Not found");  
 }  
  
 else if(ch == 2) {  
 Arrays.*sort*(*books*, Comparator.*comparing*(b -> b.title.toLowerCase()));  
 System.*out*.print("Enter title to search: ");  
 String key = sc.nextLine();  
 int low = 0, high = *books*.length - 1;  
 boolean found = false;  
 while(low <= high) {  
 int mid = (low + high) / 2;  
 int cmp = *books*[mid].title.compareToIgnoreCase(key);  
 if(cmp == 0) {  
 System.*out*.println("Found: " + *books*[mid]);  
 found = true;  
 break;  
 }  
 else if(cmp < 0) low = mid + 1;  
 else high = mid - 1;  
 }  
 if(!found) System.*out*.println("Not found");  
 }  
  
 else if(ch == 3) break;  
  
 else System.*out*.println("Invalid option");  
 }  
 }  
}

**Output:**

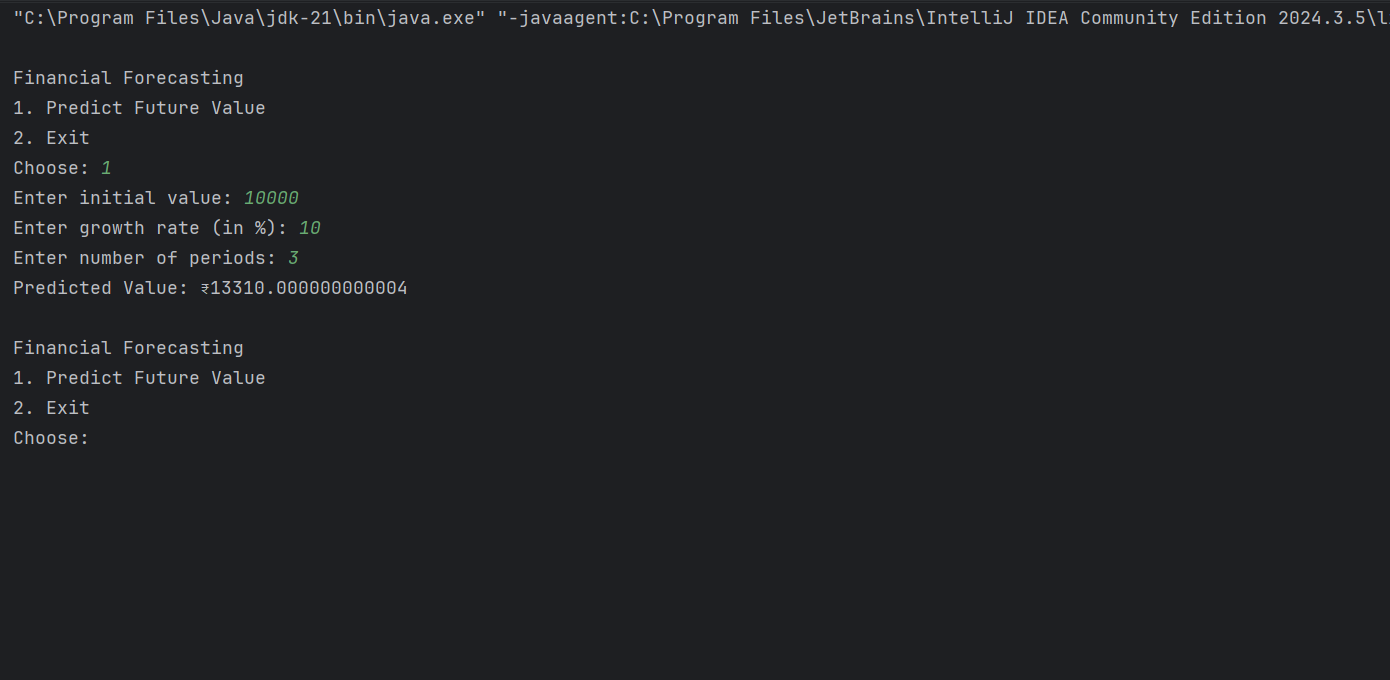
****

**Exercise 7: Financial Forecasting (Recursive Growth Prediction)**

**Code:**

import java.util.\*;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
  
 while(true) {  
 System.*out*.println("\nFinancial Forecasting");  
 System.*out*.println("1. Predict Future Value");  
 System.*out*.println("2. Exit");  
 System.*out*.print("Choose: ");  
 int ch = sc.nextInt();  
  
 if(ch == 1) {  
 System.*out*.print("Enter initial value: ");  
 double amount = sc.nextDouble();  
 System.*out*.print("Enter growth rate (in %): ");  
 double rate = sc.nextDouble();  
 System.*out*.print("Enter number of periods: ");  
 int n = sc.nextInt();  
  
 double future = *futureValue*(amount, rate, n);  
 System.*out*.println("Predicted Value: ₹" + future);  
 }  
  
 else if(ch == 2) break;  
  
 else System.*out*.println("Invalid choice");  
 }  
 }  
  
 static double futureValue(double amount, double rate, int n) {  
 if(n == 0) return amount;  
 return *futureValue*(amount \* (1 + rate / 100), rate, n - 1);  
 }  
}

**Output:**

****