**1. Reverse Queue Using Stack**

**//Code**

**import java.util.\*;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**Queue<String> queue = new LinkedList<>();**

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

**int n = sc.nextInt();**

**sc.nextLine();**

**for (int i = 1; i <= n; i++) {**

**System.out.print("Enter name of customer " + i + ": ");**

**String name = sc.nextLine();**

**queue.add(name);**

**}**

**System.out.println("\nOriginal Queue: " + queue);**

**Stack<String> stack = new Stack<>();**

**while (!queue.isEmpty()) {**

**stack.push(queue.remove());**

**}**

**while (!stack.isEmpty()) {**

**queue.add(stack.pop());**

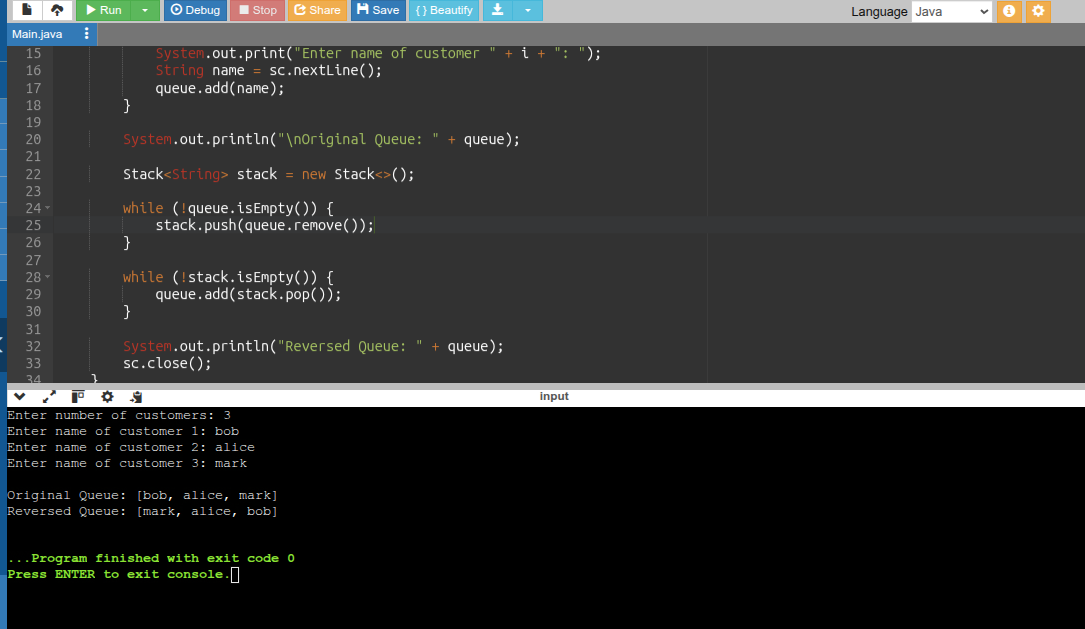
**}**

**System.out.println("Reversed Queue: " + queue);**

**sc.close();**

**}**

**}**



**2.Library Book Stack**

**//code**

**import java.util.\*;**

**public class LibraryBookStack {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**Stack<String> bookStack = new Stack<>();**

**int choice;**

**do {**

**System.out.println("\n=== Library Book Stack ===");**

**System.out.println("1. Add Book (Push)");**

**System.out.println("2. Remove Book (Pop)");**

**System.out.println("3. Peek Top Book");**

**System.out.println("4. Display All Books");**

**System.out.println("5. Exit");**

**System.out.print("Enter your choice: ");**

**choice = sc.nextInt();**

**sc.nextLine(); // Consume newline**

**switch (choice) {**

**case 1:**

**System.out.print("Enter book name to add: ");**

**String book = sc.nextLine();**

**bookStack.push(book);**

**System.out.println("Book \"" + book + "\" added to the stack.");**

**break;**

**case 2:**

**if (bookStack.isEmpty()) {**

**System.out.println("Stack is empty! No book to remove.");**

**} else {**

**String removedBook = bookStack.pop();**

**System.out.println("Removed book: " + removedBook);**

**}**

**break;**

**case 3:**

**if (bookStack.isEmpty()) {**

**System.out.println("Stack is empty! No top book.");**

**} else {**

**System.out.println("Top book: " + bookStack.peek());**

**}**

**break;**

**case 4:**

**if (bookStack.isEmpty()) {**

**System.out.println("No books in the stack.");**

**} else {**

**System.out.println("Books in stack: " + bookStack);**

**}**

**break;**

**case 5:**

**System.out.println("Exiting... Thank you!");**

**break;**

**default:**

**System.out.println("Invalid choice! Try again.");**

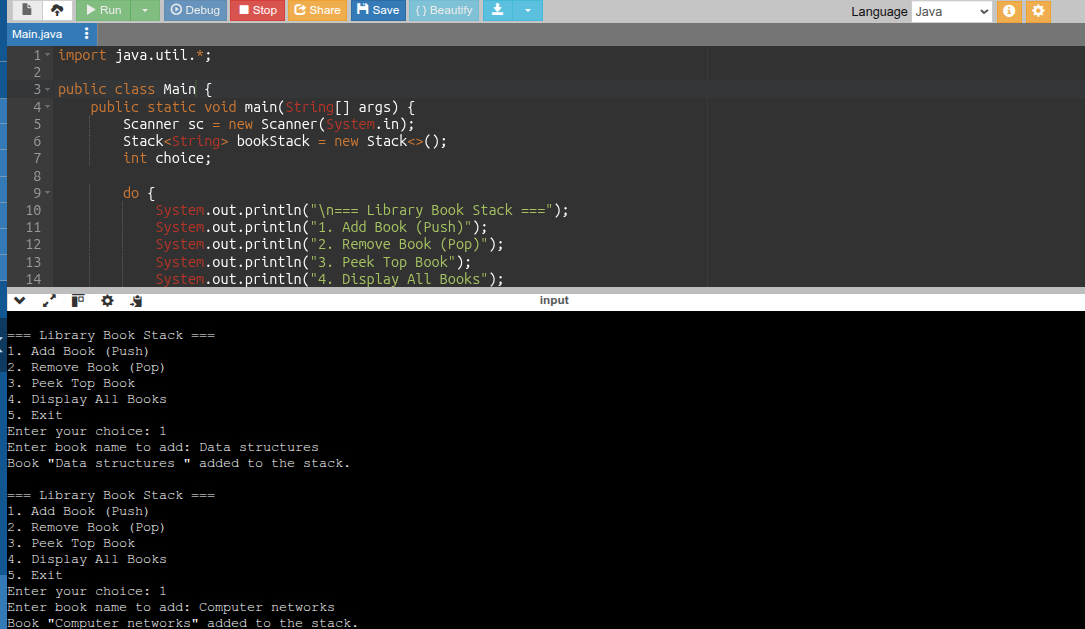
**}**

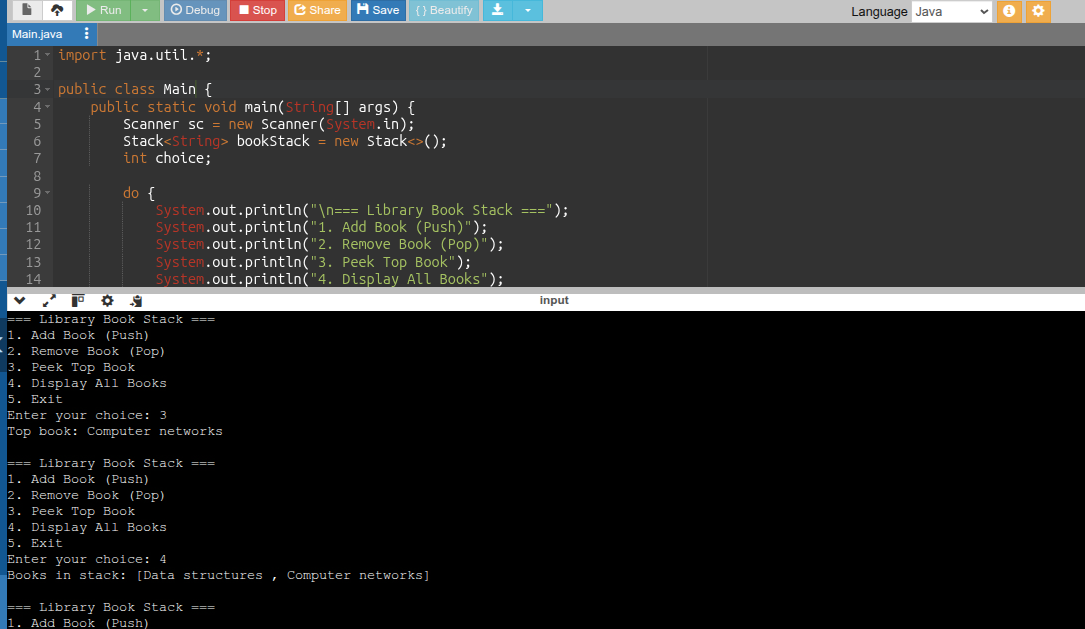
**} while (choice != 5);**

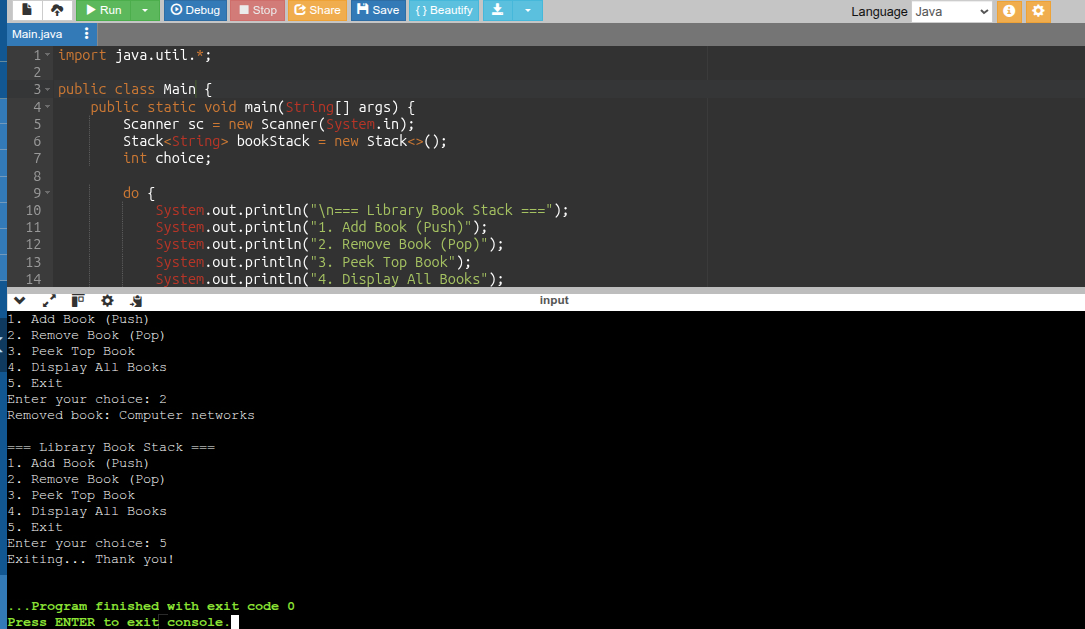
**sc.close();**

**}**

**}**





****

**3. Car wash service Queue**

**//code**

**import java.util.\*;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**LinkedList<String> carQueue = new LinkedList<>();**

**int choice;**

**do {**

**System.out.println("\n=== Car Wash Queue ===");**

**System.out.println("1. Add Normal Car (End)");**

**System.out.println("2. Add VIP Car (Front)");**

**System.out.println("3. Remove Car After Washing");**

**System.out.println("4. Display Queue");**

**System.out.println("5. Exit");**

**System.out.print("Enter your choice: ");**

**choice = sc.nextInt();**

**sc.nextLine(); // consume newline**

**switch (choice) {**

**case 1:**

**System.out.print("Enter normal car name/number: ");**

**String normalCar = sc.nextLine();**

**carQueue.addLast(normalCar);**

**System.out.println("Normal car added to the end.");**

**break;**

**case 2:**

**System.out.print("Enter VIP car name/number: ");**

**String vipCar = sc.nextLine();**

**carQueue.addFirst(vipCar);**

**System.out.println("VIP car added to the front.");**

**break;**

**case 3:**

**if (carQueue.isEmpty()) {**

**System.out.println("No cars in the queue.");**

**} else {**

**String washedCar = carQueue.removeFirst();**

**System.out.println("Car washed and removed: " + washedCar);**

**}**

**break;**

**case 4:**

**if (carQueue.isEmpty()) {**

**System.out.println("No cars waiting.");**

**} else {**

**System.out.println("Current Queue: " + carQueue);**

**}**

**break;**

**case 5:**

**System.out.println("Exiting... Thank you!");**

**break;**

**default:**

**System.out.println("Invalid choice. Please try again.");**

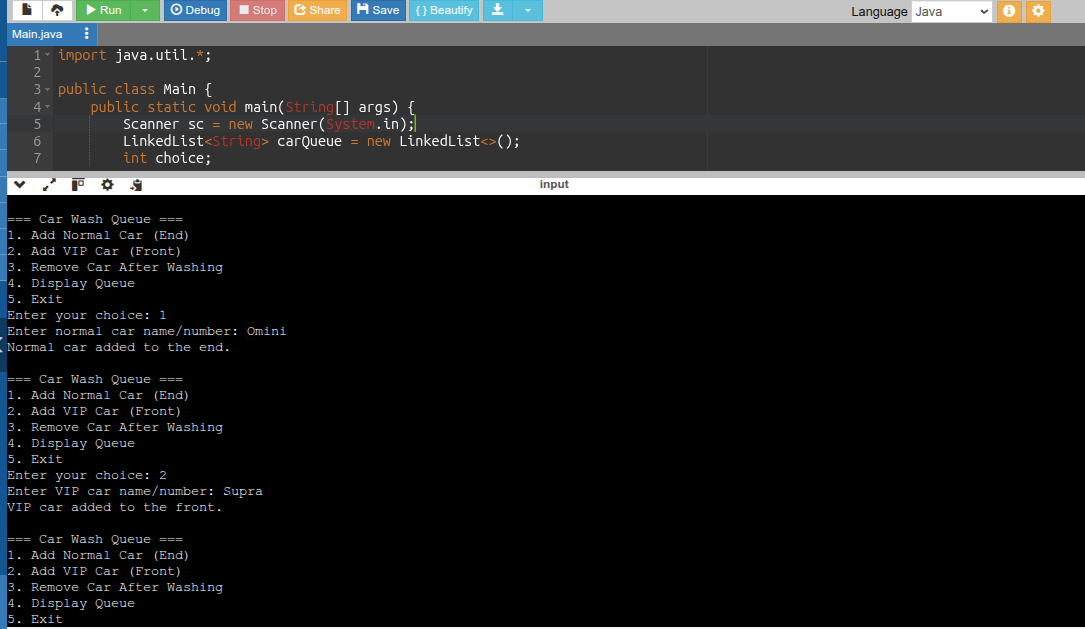
**}**

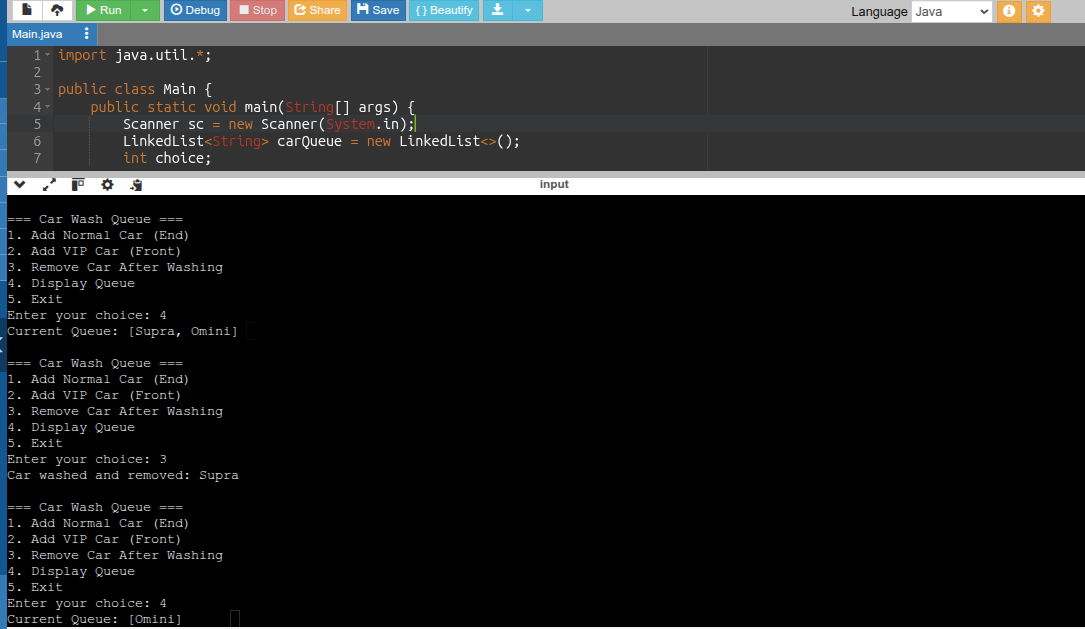
**} while (choice != 5);**

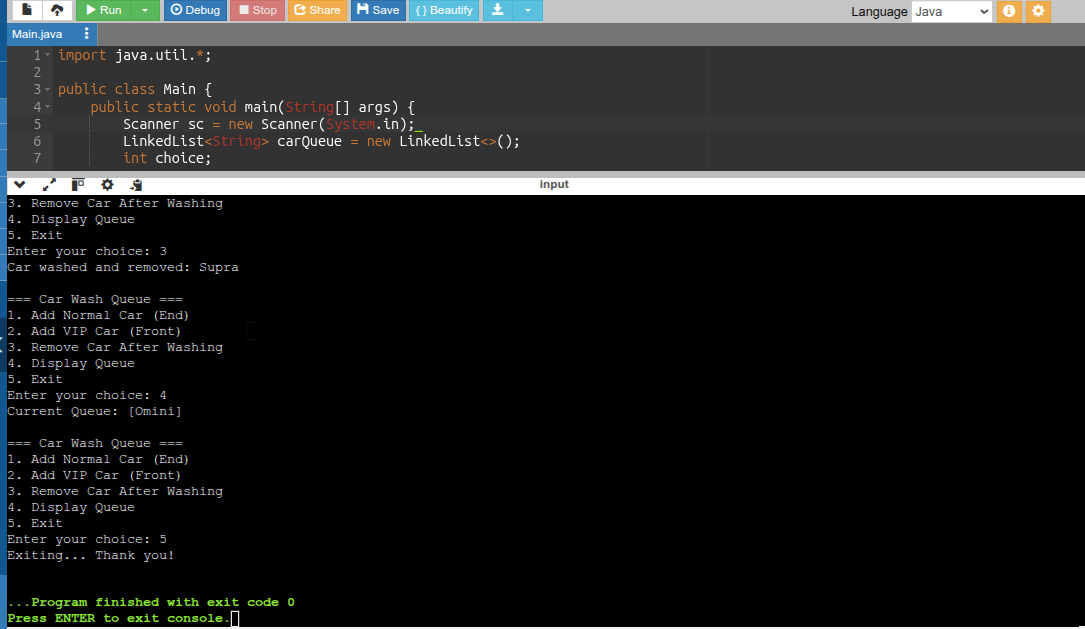
**sc.close();**

**}**

**}**







**4.Undo-Redo Function (using stack)**

**//code**

**import java.util.\*;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**Stack<String> undoStack = new Stack<>();**

**Stack<String> redoStack = new Stack<>();**

**int choice;**

**do {**

**System.out.println("\n=== Undo-Redo System ===");**

**System.out.println("1. Perform Action");**

**System.out.println("2. Undo");**

**System.out.println("3. Redo");**

**System.out.println("4. Show Current State");**

**System.out.println("5. Exit");**

**System.out.print("Enter your choice: ");**

**choice = sc.nextInt();**

**sc.nextLine(); // consume newline**

**switch (choice) {**

**case 1:**

**System.out.print("Enter action to perform: ");**

**String action = sc.nextLine();**

**undoStack.push(action);**

**redoStack.clear(); // clear redo stack after new action**

**System.out.println("Action performed: " + action);**

**break;**

**case 2:**

**if (undoStack.isEmpty()) {**

**System.out.println("Nothing to undo!");**

**} else {**

**String undoneAction = undoStack.pop();**

**redoStack.push(undoneAction);**

**System.out.println("Undid action: " + undoneAction);**

**}**

**break;**

**case 3:**

**if (redoStack.isEmpty()) {**

**System.out.println("Nothing to redo!");**

**} else {**

**String redoneAction = redoStack.pop();**

**undoStack.push(redoneAction);**

**System.out.println("Redid action: " + redoneAction);**

**}**

**break;**

**case 4:**

**System.out.println("Undo Stack: " + undoStack);**

**System.out.println("Redo Stack: " + redoStack);**

**break;**

**case 5:**

**System.out.println("Exiting... Thank you!");**

**break;**

**default:**

**System.out.println("Invalid choice! Try again.");**

**}**

**} while (choice != 5);**

**sc.close();**

**}**

**}**

