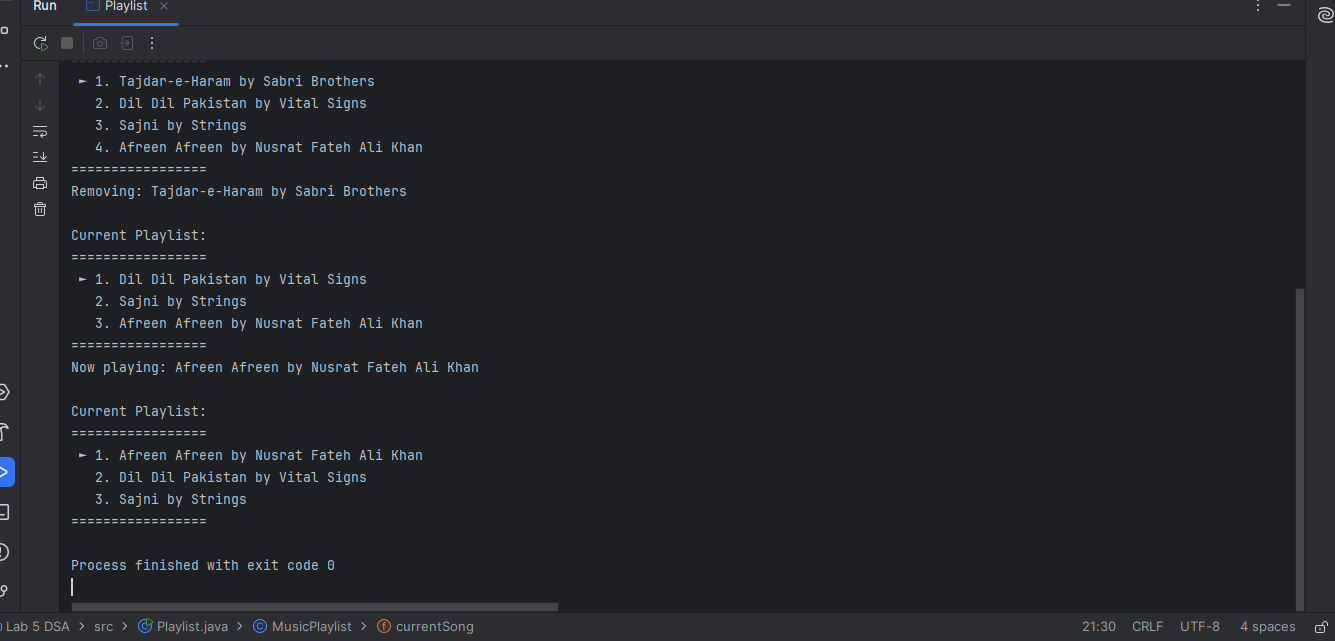
**Play List**

class Song {  
 String title;  
 String artist;  
 Song next;  
 Song prev;  
  
 public Song(String title, String artist) {  
 this.title = title;  
 this.artist = artist;  
 this.next = null;  
 this.prev = null;  
 }  
  
 @Override  
 public String toString() {  
 return title + " by " + artist;  
 }  
}  
  
class MusicPlaylist {  
 private Song currentSong;  
 private int size;  
  
 public MusicPlaylist() {  
 this.currentSong = null;  
 this.size = 0;  
 }  
  
 public void addSong(String title, String artist) {  
 Song newSong = new Song(title, artist);  
  
 if (currentSong == null) {  
 currentSong = newSong;  
  
 newSong.next = newSong;  
 newSong.prev = newSong;  
 }  
 else {  
  
 newSong.next = currentSong.next;  
 newSong.prev = currentSong;  
 currentSong.next.prev = newSong;  
 currentSong.next = newSong;  
  
 currentSong = newSong;  
 }  
 size++;  
 System.*out*.println("Added: " + title + " by " + artist);  
 }  
  
 public Song playNext() {  
 if (currentSong == null) {  
 System.*out*.println("Playlist is empty");  
 return null;  
 }  
  
 currentSong = currentSong.next;  
 System.*out*.println("Now playing: " + currentSong);  
 return currentSong;  
 }  
  
  
 public Song playPrevious() {  
 if (currentSong == null) {  
 System.*out*.println("Playlist is empty");  
 return null;  
 }  
  
 currentSong = currentSong.prev;  
 System.*out*.println("Now playing: " + currentSong);  
 return currentSong;  
 }  
  
 public void removeSong() {  
 if (currentSong == null) {  
 System.*out*.println("Playlist is empty");  
 return;  
 }  
  
 System.*out*.println("Removing: " + currentSong);  
  
 if (currentSong.next == currentSong) {  
 currentSong = null;  
 } else {  
 Song nextSong = currentSong.next;  
 currentSong.prev.next = currentSong.next;  
 currentSong.next.prev = currentSong.prev;  
 currentSong = nextSong;  
 }  
 size--;  
 }  
  
 // Display the current playlist  
 public void displayPlaylist() {  
 if (currentSong == null) {  
 System.*out*.println("Playlist is empty");  
 return;  
 }  
  
 System.*out*.println("\nCurrent Playlist:");  
 System.*out*.println("=================");  
  
 Song temp = currentSong;  
 int count = 0;  
  
 do {  
 String currentMarker = (temp == currentSong) ? " ► " : " ";  
 System.*out*.println(currentMarker + (count + 1) + ". " + temp);  
 temp = temp.next;  
 count++;  
 } while (temp != currentSong);  
  
 System.*out*.println("=================");  
 }  
  
  
 public int getSize() {  
 return size;  
 }  
  
  
 public Song getCurrentSong() {  
 return currentSong;  
 }  
}  
  
public class Playlist {  
 public static void main(String[] args) {  
 MusicPlaylist playlist = new MusicPlaylist();  
  
  
 playlist.addSong("Afreen Afreen", "Nusrat Fateh Ali Khan");  
 playlist.addSong("Tajdar-e-Haram", "Sabri Brothers");  
 playlist.addSong("Dil Dil Pakistan", "Vital Signs");  
 playlist.addSong("Sajni", "Strings");  
  
  
  
 playlist.displayPlaylist();  
  
  
 playlist.playNext();  
 playlist.playNext();  
  
  
 playlist.displayPlaylist();  
  
 playlist.removeSong();  
  
  
 playlist.displayPlaylist();  
  
  
 playlist.playPrevious();  
 playlist.displayPlaylist();  
 }  
}



**Text Editor**

import java.util.Stack;  
  
class TextAction {  
 enum ActionType {  
 *INSERT*,  
 *DELETE* }  
  
 ActionType type;  
 String content;  
 int position;  
  
 public TextAction(ActionType type, String content, int position) {  
 this.type = type;  
 this.content = content;  
 this.position = position;  
 }  
  
 @Override  
 public String toString() {  
 return type + " \"" + content + "\" at position " + position;  
 }  
}  
  
class TextEditor {  
 private String text;  
 private Stack<TextAction> actionStack;  
  
 public TextEditor() {  
 text = "";  
 actionStack = new Stack<>();  
 }  
  
 public void insert(String content, int position) {  
 if (position < 0 || position > text.length()) {  
 System.*out*.println("Invalid position");  
 return;  
 }  
  
 String beforeInsert = text.substring(0, position);  
 String afterInsert = text.substring(position);  
 text = beforeInsert + content + afterInsert;  
  
 TextAction action = new TextAction(TextAction.ActionType.*INSERT*, content, position);  
 actionStack.push(action);  
  
 System.*out*.println("Inserted: \"" + content + "\" at position " + position);  
 System.*out*.println("Current text: " + text);  
 }  
  
 public void delete(int position, int length) {  
 if (position < 0 || position + length > text.length()) {  
 System.*out*.println("Invalid position or length");  
 return;  
 }  
  
 String deletedContent = text.substring(position, position + length);  
  
 String beforeDelete = text.substring(0, position);  
 String afterDelete = text.substring(position + length);  
 text = beforeDelete + afterDelete;  
  
 TextAction action = new TextAction(TextAction.ActionType.*DELETE*, deletedContent, position);  
 actionStack.push(action);  
  
 System.*out*.println("Deleted: \"" + deletedContent + "\" from position " + position);  
 System.*out*.println("Current text: " + text);  
 }  
  
 public void undo() {  
 if (actionStack.isEmpty()) {  
 System.*out*.println("Nothing to undo");  
 return;  
 }  
  
 TextAction lastAction = actionStack.pop();  
  
 if (lastAction.type == TextAction.ActionType.*INSERT*) {  
 String beforeDelete = text.substring(0, lastAction.position);  
 String afterDelete = text.substring(lastAction.position + lastAction.content.length());  
 text = beforeDelete + afterDelete;  
 System.*out*.println("Undid insertion: \"" + lastAction.content + "\"");  
 } else if (lastAction.type == TextAction.ActionType.*DELETE*) {  
 String beforeInsert = text.substring(0, lastAction.position);  
 String afterInsert = text.substring(lastAction.position);  
 text = beforeInsert + lastAction.content + afterInsert;  
 System.*out*.println("Undid deletion: \"" + lastAction.content + "\"");  
 }  
  
 System.*out*.println("Current text: " + text);  
 }  
  
 public String peekCurrentState() {  
 return text;  
 }  
  
 public void displayAllActions() {  
 if (actionStack.isEmpty()) {  
 System.*out*.println("No actions performed yet");  
 return;  
 }  
  
 System.*out*.println("\nAction History (most recent at top):");  
 System.*out*.println("====================================");  
  
 Stack<TextAction> tempStack = new Stack<>();  
 while (!actionStack.isEmpty()) {  
 tempStack.push(actionStack.pop());  
 }  
  
 int actionNumber = 1;  
 while (!tempStack.isEmpty()) {  
 TextAction action = tempStack.pop();  
 System.*out*.println(actionNumber + ". " + action);  
 actionStack.push(action);  
 actionNumber++;  
 }  
  
 System.*out*.println("====================================");  
 }  
}  
  
public class TextEditorDemo {  
 public static void main(String[] args) {  
 TextEditor editor = new TextEditor();  
  
 editor.insert("Hello", 0);  
 editor.insert(" world", 5);  
 editor.insert("!", 11);  
  
 editor.displayAllActions();  
  
 editor.undo();  
  
 editor.insert(" Java", 10);  
  
 System.*out*.println("\nCurrent state: " + editor.peekCurrentState());  
  
 editor.delete(0, 6);  
  
 editor.displayAllActions();  
  
 editor.undo();  
 editor.undo();  
  
 System.*out*.println("\nFinal state: " + editor.peekCurrentState());  
 }  
}

