**> Java implementation of an in-memory store for recently played songs:**

import java.util.\*;

public class RecentlyPlayedSongsStore {

private int capacity;

private Map<String, Deque<String>> songsByUser;

public RecentlyPlayedSongsStore(int capacity) {

this.capacity = capacity;

this.songsByUser = new HashMap<>();

}

public void addSong(String user, String song) {

Deque<String> songs = songsByUser.get(user);

if (songs == null) {

songs = new LinkedList<>();

songsByUser.put(user, songs);

}

songs.offer(song);

if (songs.size() > capacity) {

songs.poll();

}

}

public List<String> getRecentlyPlayedSongs(String user) {

Deque<String> songs = songsByUser.get(user);

if (songs == null) {

return Collections.emptyList();

} else {

return new ArrayList<>(songs);

}

}

}

> usage of the **RecentlyPlayedSongsStore** class:

RecentlyPlayedSongsStore store = new RecentlyPlayedSongsStore(3);

store.addSong("Alice", "S1");

store.addSong("Alice", "S2");

store.addSong("Alice", "S3");

System.out.println(store.getRecentlyPlayedSongs("Alice")); // prints [S1, S2, S3]

store.addSong("Alice", "S4");

System.out.println(store.getRecentlyPlayedSongs("Alice")); // prints [S2, S3, S4]

store.addSong("Alice", "S2");

System.out.println(store.getRecentlyPlayedSongs("Alice")); // prints [S3, S4, S2]

store.addSong("Alice", "S1");

System.out.println(store.getRecentlyPlayedSongs("Alice")); // prints [S4, S2, S1]