Ashton Jane
CS214 Final Project
4/29/2024

Motivation:

In many of the popular music apps, such as Apple Music and Spotify, there are features implemented that recommend the user songs to listen to based on what they "like", or the music genres, artists, and songs they listen to the most. Our program has built itself in a similar fashion, but instead of listening statistics, we use ratings. I decided to implement a similar system, that could even be expanded upon to create a user platform, where a request to see recommended songs would be correlated to that user's data profile stored in the server, and a list of songs could be sent back unique to that user.

Overview:

Building on PA6, I plan to add new functionality that will recommend new songs to the user. The program will take 1 additional input, with the same inputs as PA6, excluding the input to access the nth cluster(5 total). This additional input is the index of the desired user to find recommended songs for (index of 1 means first user, etc). The program will then go through said user's inputted ratings to find the highest rated song. If there is a tie, it will be broken by whichever song is rated higher by other users (highest average rating).

Once the highest rated song is found by user n, the cluster containing this song will be located, and a list will be generated containing the first 20 (or less) songs that are closest in Euclidian Distance to the highest rated song, that aren't that highest rated song. This will be done using a Data Structure Similar to a Queue, where each node will contain its Euclidian Distance from the highest rated song, as well as the Song Name. For Every Song in the found cluster, the distance will be calculated, and from that calculation, it will be placed in said queue. The maximum length of this queue is 20, so any values that would make it larger are removed from the end, as they are the farthest away in euclidean distance out of the 21 songs.

Once the cluster has been iterated through, an output file is constructed of the recommended songs from this queue, with the closest distance being at the top, and the 20th closest being at the bottom.