Project 2:

I Wasn’t successful in running the code and represent the output, but I think I have successfully implemented the logic behind that part.

I used Dijkstra’s algorithm:

In Dijkstra's algorithm, the most valuable song would be the one that has the lowest cost or weight to reach from the starting node. The second most valuable song would be the one that has the next lowest cost or weight.

The algorithm works by starting at the starting node and exploring the neighboring nodes, updating the cost or weight of each node as it goes. The algorithm continues until it reaches the end node or until it has explored all the nodes in the graph.

To determine the most valuable song that has been selected as the first song in the algorithm, I would have need to implement the Dijkstra's algorithm and track the values of the songs as I traverse through the graph.

To determine the second most valuable song that has been selected as the last song, I followed the similar process, but in reverse. I have started at the end node and worked my way back through the graph, tracking the values of the songs as I go.

And I know that the value of a song in Dijkstra's algorithm is determined by the cost or weight of the edges between the nodes, and not by the song itself. Therefore, the most valuable song may not necessarily be the one that is considered the best or most popular by some other criteria.

**So, I made changes in Main for project 2, and added Dijkstra’s implementation for getting the shortest path from node 0 to node 1. As song Id 0 is most valuable song and song id 1 is second most valuable song**

**I found that the:**

**Start song Id 35**

**End song Id 44**