Names: _____ **100 points total**

CS 2123 Programming Project 5 Fall 2019

Assignment is due at 11:59pm on December 3. Submit a digital copy of the assignment on Harvey. You may submit a lateness coupon request BEFORE the assignment is due by sending an email to cs2123f19@googlegroups.com with Subject "CS2123 Project Lateness Coupon". All other late work will receive a 10 percentage point deduction per day (including weekends), No late work is accepted beyond five days after the assignment is due.

A music playlist is an ordered list of songs. Songs can be represented as the triple (track name, artist, genre). Write Python code to compute the "edit distance" between two music playlists, that is, the minimum number of alterations required to make the playlists equivalent. Implement the function `playlist_transform`, whose behavior is specified by the following docstring:

```
def playlist_transform(s,t,compareType="Song"):
    """
    Computes the edit distance for two playlists s and t, and prints the minimal edits
    required to transform playlist s into playlist t.
    Inputs:
    s: 1st playlist (format: list of (track name, artist, genre) triples)
    t: 2nd playlist (format: list of (track name, artist, genre) triples)
    compareType: String indicating the type of comparison to make.
        "Song" (default): songs in a playlist are considered equivalent if the
        (song name, artist, genre) triples match.
        "Genre": songs in a playlist are considered equivalent if the same genre is used.
        "Artist": songs in a playlist are considered equivalent if the same artist is used.
    Output: The minimum edit distance and the minimal edits required to transform playlist
    s into playlist t.
    """
```

Your code should be modular and follow a dynamic programming strategy. You are welcome to base your own code on code used in lectures. Here is example output for invoking `playlist_transform` that transforms the playlist located at `https://secon.utulsa.edu/cs2123/blues1.csv` to the playlist located at `https://secon.utulsa.edu/cs2123/blues2.csv`, using compareType ="Song".

```
Comparing playlist similarity by song
6 edits required to turn playlist 1 into playlist 2.
Insert ['Texas flood', 'Stevie Ray Vaughan', 'Blues']
Leave ["Ain't nothing wrong with that", 'Robert Randolph and the Family Band', 'Rock'] unchanged
Leave ["Couldn't stand the weather", 'Stevie Ray Vaughan', 'Blues'] unchanged
Replace ['Power to love', 'Jimi Hendrix', 'Rock'] with ['San Francisco Bay Blues', 'Eric Clapton', 'Blues']
Leave ['Going in the right direction', 'Robert Randolph and the Family Band', 'Blues'] unchanged
Replace ['Red house', 'Jimi Hendrix', 'Blues'] with ['Purple Haze', 'Jimi Hendrix', 'Rock']
Replace ['Purple Haze', 'Jimi Hendrix', 'Rock'] with ['Red house', 'Jimi Hendrix', 'Blues']
Leave ['My way down', 'Chris Duarte Group', 'Rock'] unchanged
Replace ['The thrill is gone', 'B.B. King', 'Blues'] with ['The thrill is gone', 'Chris Duarte Group', 'Blues']
Replace ['The thrill is gone', 'Chris Duarte Group', 'Blues'] with ['The thrill is gone', 'B.B. King', 'Blues']
```

Additionally, you must include at least two of your own playlists for testing. Include the playlists and the output with your assignment (turn in both a hard and a copy on Harvey). You may download starter code from `https://secon.utulsa.edu/cs2123/code/playlist_starter.txt`, as well as make use of the provided `read_playlist` function that processes a CSV playlist and returns a list of triples.