final1.py - process Spotify Top 200 Charts (2020-2021) into:

1. year_data: dataframe for different genres sorting
2. genreDict: dictionary containing different genres and its numbers in the chart

In this file:

- function **isPop(piece)**: find all entries containing “pop” by “Genre” of dataframe
- function **isHipHop(piece)**: find all entries containing “hip hop” by “Genre” of dataframe
- function **isRap(piece)**: find all entries containing “rap” by “Genre” of dataframe
- function **isTrap(piece)**: find all entries containing “trap” by “Genre” of dataframe
- function **isKPop(piece)**: find all entries containing “k-pop” by “Genre” of dataframe
- function **isRNB(piece)**: find all entries containing “r&b” by “Genre” of dataframe
- function **isLatin(piece)**: find all entries containing “latin” by “Genre” of dataframe
- function **isRegg(piece)**: find all entries containing “reggaeton” by “Genre” of dataframe
- function **isHouse(piece)**: find all entries containing “house” by “Genre” of dataframe
- function **isDance(piece)**: find all entries containing “dance” by “Genre” of dataframe
- function **isEDM(piece)**: find all entries containing “edm” by “Genre” of dataframe
- function **isIndie(piece)**: find all entries containing “indie” by “Genre” of dataframe
- function **isPunk(piece)**: find all entries containing “punk” by “Genre” of dataframe
- function **isOther(piece)**: find all entries containing keywords that are not one of genre word listed above by “Genre” of dataframe

final2.py - process the web scraping of weekly charts of top songs from Spotify Charts into:

1. rankingGlobalDF/ rankingAuDF/ rankingCaDF/ rankingFrDF/ rankingGeDF/ rankingHkDF/ rankingInDF/ rankingJpDF/ rankingSaDF/ rankingSkDF/ rankingUkDF/ rankingUsDF: dataframe for different latest weekly top songs charts from different countries (Global, Australia, Canada, France, Germany, Hong Kong, India, Japan, South Africa, South Korea, UK, USA)

final3.py - process structured data retrieved from the API of the playlist - "Taylor Swift - The Eras Tour Setlist" into:

1. playListDF: dataframe of the playlist

In this file:

- function **getToken()**: get token access from Spotify api
- function **getAuthHeader(token)**: get authorized token

Final Project.py - process all the processed data from the python files above into:

1. searchDict: dictionary of data for each song -
   {'Song Name': {'Highest Charting Position': x,
   'Number of Times Charted': y, 'Streams': z, 'Artist': m,
   'Artist Followers': n, 'Genre': u, 'Popularity': v}
2. rankingDict: dictionary of latest weekly top 10 songs of different countries

3. orderedPlaylistDF: sorted dataframe of albums by the number of songs performed

In this file:

- function **getSongInfo(song)**: get song info (Highest Charting Position, Number of Times Charted, Streams, Artist, Artist Followers, Genre, Popularity) from searchDict by the song name
- function **twoSongComparison(songA, songB)**: create a dataframe comparing two songs information