1. Put headings to the prepressed data files generated using MySQL to create the input files.
    - Heading for user.csv
        - user_id (id of the user)
        - cnt (count of distinct book rated by a user)
        - attr (C/M/F to define Couple/Married People/Friends type of users)
    - Heading for rating.csv
        - user_id (id of the user)
        - book_id (id of the book)
        - rating (rating given to a book by a user)
    - Heading for book.csv
        - book_id
        - authors
        - year
        - title
        - language
2. You can find the input data files in "3. TF_IDF (Python)/input" folder.

3. Run the TF_IDF.py program in ""3. TF_IDF (Python)" folder.
    - This program generate the TF_IDF value for all the books columns. The process it uses can be understandable by the "Case Study 2" example of the following link: https://www.analyticsvidhya.com/blog/2015/08/beginners-guide-learn-content-based-recommender-systems/
    - The "TF_IDF.py" file also has sufficient comments to understand the step by step approach.

4. The Program generates following files:
    - TF (Term Frequency)
        - TF_title.csv (book_id, word, TF_title_val)
        - TF_authors.csv (book_id, author, TF_author_val)
        - TF_language.csv (book_id, language, TF_language_val)
        - TF_year.csv (book_id, year, TF_year_val)
    - IDF (Inverse Document Frequency)
        - IDF_title.csv (word, IDF_val)
        - IDF_authors.csv (author, IDF_val)
        - IDF_language.csv (language, IDF_val)
        - IDF_year.csv (year, IDF_val)
    - UA (Users Attribute or Profile for each term)
        - UA_title.csv (user_id word, UA_title_val)
        - UA_authors.csv (user_id, author, UA_author_val)
        - UA_language.csv (user_id, language, UA_language_val)
        - UA_year.csv (user_id, year, UA_year_val)
5. You can find the output files in "3. TF_IDF (Python)/output" folder.