**Code Exercise:**

**Problem Statement:** Given a GitHub repository path and source file path, find all similar code in GitHub repository directory.

**Assumption:**

1. Repository path is file path in the local computer/Operating System, and it is not the URL on the GitHub.
2. Similarity is found out based on the longest contiguous matching subsequence (LCS) that contains no “junk” elements. This does not yield minimal edit sequences but does tend to yield matches that “look right” to people. Other algorithm that can be used for fuzzy matching as well as minimal edit sequences are *edit distance or levenshtein* distance.
3. The code work on fuzzy matching instead of exact match of words and lines. This problem could be overcome by using *Kadane’s algorithms/brute force algorithms* to match exact same line of text to find copy in another file.

Workflow:

The code consists of three file important file i.e. utils.py, SquenceMatcher.py and main.py.

The utils.py contains helper functions. Functions are *keep\_only\_alphabet()*, *to\_lower()*, *replace\_character()* and *preprocessing\_operation()*. The *keep\_only\_alphabet()* function takes contents and returns content with content that only alphabets(remove everything other than alphabets). The to\_lower*()* function return the content after lowering capital letter alphabet letters. The preprocessing\_operation*()* function takes content and return content after apply preprocessing steps given by the user.

The sequence\_matcher.py file contains sequence\_matcher\_similarity*()* function. It returns similarity percentage score between source file content and GitHub file contents.

The main.py is the main python file which import other modules. It takes input from the user and gives all the file whose similarity percentage score is more than 50%. It scans the GitHub repository folder directory in breadth first search fashion.

Running Instructions:

Open CMD and run the main.py using python main.py.