A. How to invoke method:

There is 10 approximate matching methods in the code, written in Python2.7.
The first 4 methods are basic approximate matching methods and rest 6 methods are extension methods.

Methods:

```
# ----------------------------------------------------------------------------------------------------------------------------
# Basic approximate methods:
# ----------------------------------------------------------------------------------------------------------------------------
    1. Self-Write Global Edit Distance
       Invoke method name: selfwrite_GED( )

    2. System Global Edit Distance
       Invoke method name: system_GED( )

    3. Self-Write Local Edit Distance
       Invoke method name: selfwrite_LED( )

    4. Self-Write N-Gram Distance (N = 2)
       Invoke method name: selfwrite_2Gram( )

# ----------------------------------------------------------------------------------------------------------------------------
# Use exist substitution matrix:
# ----------------------------------------------------------------------------------------------------------------------------
    5. Self-Write Blosum62 Substitution Matrix + Self-Write GED
       Invoke method name: selfwrite_blosum62_GED( )

# ----------------------------------------------------------------------------------------------------------------------------
# Use self-trained matrix:
# ----------------------------------------------------------------------------------------------------------------------------
    6. Self-Write    100lines trained Matrix + Self-Write GED
        # Randomly chose 100 lines 10 times from train.txt (totally    1000 lines) for training
       Invoke method name: selfwrite_trainedMatrix_GED( improvedMatrix_100_Modified,      100 )
    7. Self-Write    500lines trained Matrix + Self-Write GED
       # Randomly chose 500 lines 10 times from train.txt (totally    5000 lines) for training
        Invoke method name: selfwrite_trainedMatrix_GED( improvedMatrix_500_Modified,      500 )

    8. Self-Write 1000lines trained Matrix + Self-Write GED
      # Randomly chose 1000 lines 10 times from train.txt (totally 10000 lines) for training
       Invoke method name: selfwrite_trainedMatrix_GED( improvedMatrix_1000_Modified, 1000 )

    Note: Method 6/7/8 invoke the same selfwrite_trainedMatrix_GED( argu1, argu2 ) method.
                argu1: The trained matrix
                argu2: Set which method to use ( This just for if to judge which file to output the results )

# ------------------------------------------------------------------------------------------------------------------------------------
# Use self-trained matrix + multiple predictions:
# ------------------------------------------------------------------------------------------------------------------------------------
    9. Self-Write 1000lines trained Matrix + Self-Write GED + Multiple predictions
          Invoke method name: selfwrite_trainedMatrix_GED_Multi( improvedMatrix_1000_Modified )

          Note: selfwrite_trainedMatrix_GED_Multi( argu )
                       argu: The trained matrix

   10. Self-Write 1000lines trained Matrix + Self-Write GED + Multiple prediction + Soundex
          Invoke method name: selfwrite_trainedMatrix_GED_Multi_Soundex( improvedMatrix_1000_Modified )

          Note: 1. selfwrite_trainedMatrix_GED_Multi_Soundex(( argu )
                             argu: The trained matrix
                   2. Not implement Soundex algorithm, just use the concept "keep the first letter" in Soundex
                       for helping improving the performance
```

B. Location and Format of the output:

B.1. Location:

The result file will be in "/home/subjects/comp90049/submission/aol3" my home directory.

```
# ---------------------------------------------------------------------------------------------------------------------------------------------
# Each method result file name will be as below:
# ---------------------------------------------------------------------------------------------------------------------------------------------
```

1. results_global_myself.txt
2. results_global_system.txt
3. results_local_myself.txt
4. results_N-Gram.txt
5. results_Blosum62Matrix.txt
6. results_100TrainedLinesMatrix.txt
7. results_500TrainedLinesMatrix.txt
8. results_1000TrainedLinesMatrix.txt
9. results_1000TrainedLinesMatrix_Multi.txt
10. results_1000TrainedLinesMatrix_Multi_Soundex.txt

B.2. Format of the output:

```
# ---------------------------------------------------------------------------------------------------------------------------------------------
# The first 8 result files will be in same format (Perdict 1 name):
# ---------------------------------------------------------------------------------------------------------------------------------------------
   Format1: PERSIAN-NAME     \t     latin-name       \n

     For example:
          AACTAY   actaeon
          AALTJ      aaltje
          AAMYNA  aamina
          AARVN     aaron
          ABA    aba
          ABAD abad
          ABADA     awadia
          .....
```

```
# ---------------------------------------------------------------------------------------------------------------------------------------
# The last 2 result files will be in same format (Perdict multiple names):
# ---------------------------------------------------------------------------------------------------------------------------------------

   Format2: PERSIAN-NAME    \t    latin-name1   ' '   latin-name2   ' '   latin-name3   ......   latin-nameN    \n

   Note:
        1. if just one latin name get the best score, the output will be like Format1.
        2. ' ' in the Formate2 means blankspace.

    For example:

        AACTAY   actaeon
        AALTJ    aaltje
        AAMYNA   aamina
        AARVN    aaron
        ABA    aba
        ABAD abad
        ABADA    awadia
        ABADHBY     abudhabi
        ABAJA    abuja
        ABAR aabar   akbar   anbar
        ABCR abcher
        ABDJA    abidjan
        ABDLRHMAN abdulrahman
        ABDVN    aberdeen
        ABH    abe abhay
        ABHAY    abhay
        ABHY abhay
        ABLA abella
        ABL    abel    able    abul
        ABL    abel    able    abul
        ABLH abilock able
        .....
```