The purpose of the task is to write a program that clusters characters.

Input to the program is the following:

- A file provided as a command-line argument, that contains in each line the name (and possibly the path) of the image file (example).
- Image files listed in the above file

The output from the program should be:

- A file containing in each line **a space-separated** list of files containing images belonging to the given cluster. The ends of the lines in the unix convention. The output file should contain only the files' names of the images that belong to the same cluster which are space separated & **without any additional text related to the directory** (example).
- File in html format that displays images belonging to individual clusters. Clusters should be separated using horizontal lines ( `<HR>` ).

Task specifics:

- Unknown number of decision classes - in addition to regular letters, there will be fragments of characters and characters glued together.
- Some letters may remain indistinguishable, eg "I", "l" and "1" or "m" and "rn".
- When designing a metric, keep in mind that the characters remain similar if they are shifted on the pictures, for example: and

The task evaluation will depend on the effectiveness of the selected clustering method. Effectiveness of the method will be calculated by comparing the clustering results of the test sample with optimal clustering, which is done manually, using Rand index. Test sample contains approximately 5000 characters.
Sample of approximately 7600 characters

Article describing classic clustering methods

The solution should consist of:

1. The source code of the program

2. A file containing all necessary commands that prepares the execution environment, installs all required libraries and runs the code. All codes will be executed by creating a separate python virtual environment. So, please provide the commands required to create and activate python virtual environment and then install the required libraries (example).

Note: It is also possible to submit your solution in R. Please remember to provide the detailed steps to run your code.

3. Short README File with description of method, expected duration of execution and custom options. It is also necessary to add in this file a section called How to Run.