Regular Expression Assignments

1. Haigy Paigy

Haigy Paigy is as a children's invented language which sounds exactly like English, except that “aig” is inserted before the vowel sound of each syllable. E.g., the English word “hello” becomes “haigellaigo” in Haigy Paigy. Write a set of regular expressions that will automatically translate sentences from English into Haigy Paigy. The input of your program will be a string containing one English sentence. Your regular expressions should translate this sentence into Haigy Paigy.

Simplifications: only vowels can be syllable nuclei; "y" is always a vowel; several consecutive vowel letters always make a single vowel sound; the final "e" is always silent, except when it is the only vowel letter in the word.

Example input: This is a test.

Example output: Thaigis aigis aiga taigest.

2. Article Replacement

Write a set of regular expressions that will automatically replace all instances of the indefinite article ‘a’/’an’ with the definite article ‘the’ and vice versa. The input of your program will be a string of English words.

Example input: The student. An example.

Example output: A student. The example.

3. Turkish Plurals

As many other agglutinative languages, Turkish has a feature called vowel harmony. Turkish has eight vowels that are divided into two groups as follows:
the back vowels are A I O U
the front vowels are E İ Ö Ü

Note: Iı and İi are different vowels!

There is a single plural suffix in Turkish, but it has two variants (allomorphs): -ler and -lar. The vowel in the suffix mirrors the last vowel of its noun. We must add -lar to words whose final vowel is any of the back vowels, and -ler to words whose final vowel is any of the front vowels.

Examples:

- balta 'axe' : baltalar 'axes'

- kapı 'door' : kapılar 'doors'
- palto 'overcoat' : paltolar 'overcoats'
- boncuk 'bead' : boncuklar 'beads'
- ev 'house' : evler 'houses'
- kedi 'cat' : kediler 'cats'
- göz 'eye' : gözler 'eyes'
- ödül 'award' : ödüller 'awards'

Important: For simplicity and in order to avoid dealing with Unicode, we will encode ö and ü as the upper-case letters O and U, respectfully. Also, we will encode the undotted ı as the upper-case I. For example, we will write **gözler** as **gOzler**, **ödül** as **OdUl**, and **kapı** as **kapI**.

Write a set of regular expressions that will pluralize Turkish nouns. The input of your program will be a singular Turkish noun. Your regular expressions should make this noun plural.

Example input: gOz

Example output: gOzler

4. Corpus Cleaning

Linguists often need to use computers to deal with human language, especially when the dataset is very large. Large collections of authentic language are called *corpora* (singular corpus), and the field that primarily deals with analyzing corpora for patterns is called corpus linguistics.

However, computers often require special formatting of human language in order to properly analyze it. Before corpus linguists begin their analysis, they often have to first *clean* their data, often involving the removal of (certain types of) punctuation, extra whitespace, unwanted characters, etc.

Write a set of regular expressions that removes punctuation and extra whitespace from an English sentence. The input of your program will be a string containing one English sentence. Your regular expressions should remove the punctuation and extra whitespace from this sentence.

Example input: This is an example sentence, and here is another.

Example output: This is an example sentence and here is another