1. Write Python code and use MapReduct to count occurrences of each word in the first text file (file.txt). How many times each word is repeated?

```
# IMporting required libraries
from collections import defaultdict
import re
from functools import reduce

# reading file1 and converting text to lowercase
with open("file1.txt", "r", encoding="utf-8") as file:
    text1 = file.read().lower()

# extracting words and storing in word1 for counting them
words1 = re.findall(r"\b[a-zA-Z]+\b", text1)

# MAP stage - creating word-count pair each word
wd_pairs = [(word, 1) for word in words1]

# SHUFFLE & GROUP stage - we gather counts for each word and createing a dictionary with word as its key and counting its freq. and using it as value.
wd_groups = defaultdict(list)
for word, count in wd_pairs:
    wd_groups[word].append(count)

# REDUCE stage - add counts for every word
word_counts = {word: reduce(lambda x, y: x + y, counts) for word, counts in wd_groups.items()}

# Display results in alphabetical order
print("Word Count pairs in file1")
for word, count in sorted(word_counts.items()):
    print(f"{word}: {count}")
```

```
Word Count pairs in file1
a: 42
able: 2
about: 2
aching: 1
activity: 1
advanced: 1
afraid: 1
after: 2
again: 2
ah: 1
alive: 1
all: 4
allowed: 1
alone: 1
also: 1
american: 1
amounts: 1
an: 4
and: 42
angrily: 1
angry: 2
another: 1
```

```
anxiously: 1
any: 2
anyway: 1
appeared: 2
are: 3
around: 4
as: 11
ask: 2
asking: 1
at: 13
attached: 1
aunt: 5
back: 10
bad: 1
banshee: 1
bared: 1
barn: 1
bars: 6
bathroom: 1
bats: 1
be: 6
beak: 1
bearing: 1
beating: 1
bed: 6
bedroom: 2
been: 8
before: 3
behind: 1
best: 1
birds: 1
birthday: 1
blinked: 1
born: 1
bottom: 1
```

2. From the second text file (file2.txt), write Python code and use MapReduct to count how many times non-English words (names, places, spells etc.) were used. List those words and how many times each was repeated.

There are multiple ways of doing this. You can use pyenchant (https://pypi.org/project/pyenchant/), pyspellchecker (https://pyspellchecker.readthedocs.io/en/latest/) or just download a list of words (http://www.gwicks.net/dictionaries.htm) and search through them

```
import re
from collections import Counter

# load english3 file
def load_words(dictionary_file):
    with open(dictionary_file, 'r') as file:
        return {line.strip().lower() for line in file}

# reading text from file2 fro comparision
def read_text(file_path):
    with open(file_path, 'r', encoding='utf-8') as file:
        return file.read().lower()

# extracting wrods for comparision
def get_words(text):
    return re.findall(r"\b[a-zA-Z'-]+\b", text)

# checking the words for non eng words
def find_unknown_words(word_list, known_words):
    common_short_forms = {"it's", "he's", "what's", "she's", "let's", "i'm", "you're", "we're", "they're"}
    return [word for word in word_list if word not in known_words and word not in common_short_forms]

def main():
    known_words = load_words("english3.txt")
    word_list = get_words(read_text("file2.txt"))
    unknown_words = find_unknown_words(word_list, known_words)
    word_counts = Counter(unknown_words)

    # top 15 unknown eng words
    for word, count in word_counts.most_common(15):
        print(f"{word}: {count}")

if __name__ == "__main__":
    main()
```

```
weasley: 20
malfoy: 15
lockhart: 11
rowling: 8
harry's: 5
weasleys: 4
gilderoy: 4
ginny's: 4
hogwarts: 2
hagrid: 2
hagrid's: 2
ter: 2
floo: 2
me-not: 1
wizard's: 1
```

Start coding or generate with AI.