603 Assignment 2
Map Reduce Assignment

Birthday: 29- March-2000

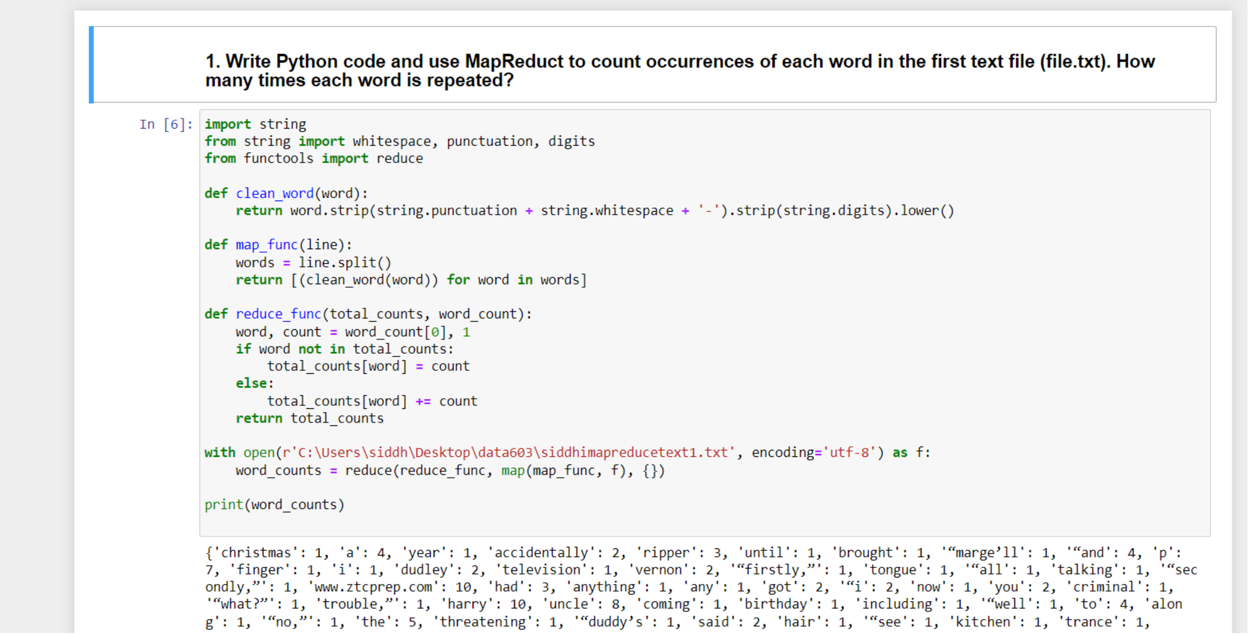
Name: Jaggari Siddhidhatri Rohit reddy
Enrollment no : LO55859

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

Code:

```
import string
from string import whitespace, punctuation, digits
from functools import reduce

def clean_word(word):
    return word.strip(string.punctuation + string.whitespace + '-').strip(string.digits).lower()

def map_func(line):
    words = line.split()
    return [(clean_word(word)) for word in words]

def reduce_func(total_counts, word_count):
    word, count = word_count[0], 1
    if word not in total_counts:
        total_counts[word] = count
    else:
        total_counts[word] += count
    return total_counts

with open(r'C:\Users\siddh\Desktop\data603\siddhimapreducetext1.txt', encoding='utf-8') as f:
    word_counts = reduce(reduce_func, map(map_func, f), {})

print(word_counts)
```

Code Snapshot :



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?

```
In [6]: import string
        from string import whitespace, punctuation, digits
        from functools import reduce

        def clean_word(word):
            return word.strip(string.punctuation + string.whitespace + '-').strip(string.digits).lower()

        def map_func(line):
            words = line.split()
            return [(clean_word(word)) for word in words]

        def reduce_func(total_counts, word_count):
            word, count = word_count[0], 1
            if word not in total_counts:
                total_counts[word] = count
            else:
                total_counts[word] += count
            return total_counts

        with open(r'C:\Users\siddh\Desktop\data603\siddhimapreducetext1.txt', encoding='utf-8') as f:
            word_counts = reduce(reduce_func, map(map_func, f), {})

        print(word_counts)
```

```
{'christmas': 1, 'a': 4, 'year': 1, 'accidentally': 2, 'ripper': 3, 'until': 1, 'brought': 1, '“marge’ll': 1, '“and': 4, 'p':
7, 'finger': 1, 'i': 1, 'dudley': 2, 'television': 1, 'vernon': 2, '“firstly,”': 1, 'tongue': 1, '“all': 1, 'talking': 1, '“sec
ondly,”': 1, 'www.ztcprep.com': 10, 'had': 3, 'anything': 1, 'any': 1, 'got': 2, '“i': 2, 'now': 1, 'you': 2, 'criminal': 1,
'“what?”': 1, 'trouble,”': 1, 'harry': 10, 'uncle': 8, 'coming': 1, 'birthday': 1, 'including': 1, '“well': 1, 'to': 4, 'alon
g': 1, '“no,”': 1, 'the': 5, 'threatening': 1, '“duddy’s': 1, 'said': 2, 'hair': 1, '“see': 1, 'kitchen': 1, 'trance': 1,
```

Output :

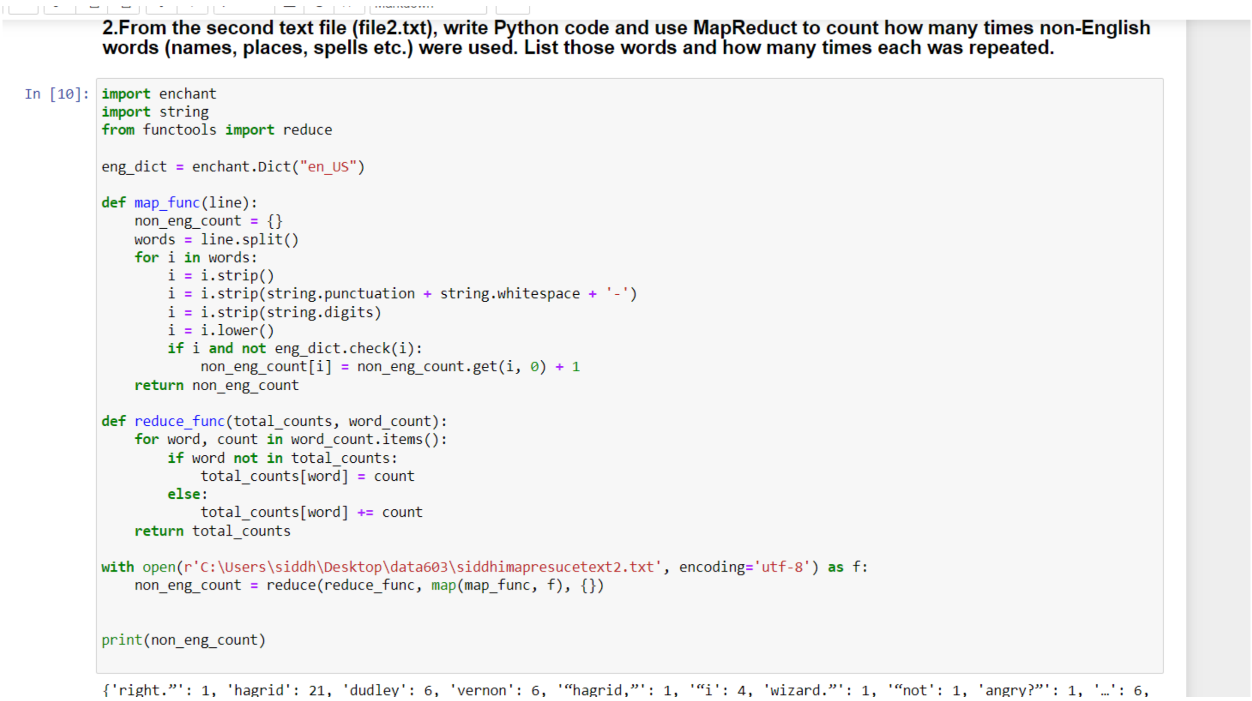
```
{'christmas': 1, 'a': 4, 'year': 1, 'accidentally': 2, 'ripper': 3, 'until': 1, 'brought': 1, '“marge’ll': 1, '“and': 4,
'p': 7, 'finger': 1, 'i': 1, 'dudley': 2, 'television': 1, 'vernon': 2, '“firstly,”': 1, 'tongue': 1, '“all': 1, 'talking': 1,
'“secondly,”': 1, 'www.ztcprep.com': 10, 'had': 3, 'anything': 1, 'any': 1, 'got': 2, '“i': 2, 'now': 1, 'you': 2,
'criminal': 1, '“what?”': 1, 'trouble,”': 1, 'harry': 10, 'uncle': 8, 'coming': 1, 'birthday': 1, 'including': 1,
'“well': 1, 'to': 4, 'along': 1, '“no,”': 1, 'the': 5, 'threatening': 1, '“duddy’s': 1, 'said': 2, 'hair': 1, '“see': 1,
'kitchen': 1, 'trance': 1, '“i’m': 1, '“like': 1, 'ask': 1, '“third': 1, 'visit': 1, '“so?”': 1, '“well,”': 1, 'be': 2, 'st': 1,
'“st': 2, 'boys!”': 1, 'voice': 1, '“exactly,”': 1, 'vernon’s': 1, 'have': 1, '“you’ll': 1, 'roared': 1, 'raised': 1,
'“knocking': 1, 'marge': 1, 'ugly': 1, '“but': 1, 'quickly': 1, 'go': 1, 'and': 2, 'over': 1, 'throbbing': 1,
'“right,”': 1, 'behavior': 1, 'of': 2, 'sign': 1, 'he': 2, 'slammed': 1, 'glass': 1, 'upstairs': 1, 'real': 1, 'cards':
1, 'his': 1, 'seemed': 1, 'asleep': 1, 'poked': 1, '“hedwig,”': 1, 'clear': 1, 'me': 1, 'reproachful': 1,
'hermione.”': 1, 'ten': 1, 'miserable': 1, 'wardrobe': 1, 'but': 1, 'time': 2, 'guest': 1, '“do': 1, 'snapped': 1,
'lie': 1, 'untidier': 1, 'all': 1, 'then': 1, 'garden': 1, '“get': 1, 'on': 1, 'even': 1, 'one': 1, 'under': 2, 'bulldog':
1, '“where’s': 1, 'my': 1, 'plastered': 1, 'suitcase': 1, 'planted': 1, 'with': 1, 'it': 1, '“petunia!”': 1, 'stand': 1,
'petunia': 3, 'large': 1, '“tea': 1, '“ripper': 1, 'aunt': 2, 'leaving': 1, 'case': 1, 'as': 1, 'by': 1, 'was': 1,
'flecked': 1, '“who’s': 1, '“oh': 1, 'boomed': 1, 'directed': 1, '“so!”': 1, '“yes,”': 1, '“don’t': 2, 'damn': 1,
'you’d': 1, 'been': 1, 'orphanage': 1, 'into': 1, 'see': 1, 'school': 1, 'took': 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.¶

Code :

```
import enchant
import string
from functools import reduce

eng_dict = enchant.Dict("en_US")

def map_func(line):
    non_eng_count = {}
    words = line.split()
    for i in words:
        i = i.strip()
        i = i.strip(string.punctuation + string.whitespace + '-')
        i = i.strip(string.digits)
        i = i.lower()
        if i and not eng_dict.check(i):
            non_eng_count[i] = non_eng_count.get(i, 0) + 1
    return non_eng_count

def reduce_func(total_counts, word_count):
    for word, count in word_count.items():
        if word not in total_counts:
            total_counts[word] = count
        else:
            total_counts[word] += count
    return total_counts

with open(r'C:\Users\siddh\Desktop\data603\siddhimapresucetext2.txt', encoding='utf-8') as f:
    non_eng_count = reduce(reduce_func, map(map_func, f), {})


print(non_eng_count)
```

Code Snapshot :



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.

```
In [10]: import enchant
         import string
         from functools import reduce

         eng_dict = enchant.Dict("en_US")

         def map_func(line):
             non_eng_count = {}
             words = line.split()
             for i in words:
                 i = i.strip()
                 i = i.strip(string.punctuation + string.whitespace + '-')
                 i = i.strip(string.digits)
                 i = i.lower()
                 if i and not eng_dict.check(i):
                     non_eng_count[i] = non_eng_count.get(i, 0) + 1
             return non_eng_count

         def reduce_func(total_counts, word_count):
             for word, count in word_count.items():
                 if word not in total_counts:
                     total_counts[word] = count
                 else:
                     total_counts[word] += count
             return total_counts

         with open(r'C:\Users\siddh\Desktop\data603\siddhimapresucetext2.txt', encoding='utf-8') as f:
             non_eng_count = reduce(reduce_func, map(map_func, f), {})


         print(non_eng_count)
```

```
{'right.”': 1, 'hagrid': 21, 'dudley': 6, 'vernon': 6, '“hagrid,”': 1, '“i': 4, 'wizard.”': 1, '“not': 1, 'angry?”': 1, '…': 6,
```

Output :

```
{'right.”': 1, 'hagrid': 21, 'dudley': 6, 'vernon': 6, '“hagrid,”': 1, '“i': 4, 'wizard.”': 1, '“not': 1, 'angry?”': 1,
'…': 6, 'www.ztcprep.com': 10, 'dudley’s': 1, '–': 7, 'j.k': 7, 'rowling': 7, '“see?”': 1, '“harry': 1, '—': 24,
'hogwarts.”': 1, '“haven’t': 1, 'going?”': 1, '“he’s': 1, 'i’ve': 2, '—”': 5, '“if': 1, 'him,”': 2, '“stop': 1, 'an’': 9,
'james': 1, 'goin’': 1, 'hogwarts': 5, 'albus': 2, 'dumbled—”': 1, 'tricks!”': 1, '“never': 1, '“—': 1,
'dumbledore': 3, 'me!”': 1, '“shouldn’ta': 1, 'temper,”': 1, '“but': 2, 'do.”': 1, '“be': 1, 'yeh': 7, 'hogwarts,”':
1, '“i’m': 2, 'speakin’': 1, 'o’': 5, '“why': 2, 'magic?”': 1, '“oh': 1, 'meself': 1, 'dumbledore.”': 1,
'expelled?”': 1, '“it’s': 1, 'gettin’': 2, 'tomorrow,”': 1, '“gotta': 1, 'that.”': 2, '“you': 1, 'that,”': 2, '“don’': 1,
'pockets.”': 2, 'diagon': 1, '“it': 1, 'dream,”': 1, 'i’ll': 2, 'cupboard.”': 1, 'there’': 1, '“all': 1, 'right,”': 1, 'up.”':
1, 'hagrid’s': 3, '“don’t': 2, '“hagrid!”': 1, '“there’s': 1, '“pay': 1, '“what?”': 1, '“he': 2, 'payin’': 1, 'deliverin’':
1, '“give': 1, 'knuts,”': 1, '“knuts?”': 1, '“the': 1, 'ones.”': 1, '“best': 1, 'london': 1, 'school.”': 1, '“um': 1,
'hagrid?”': 1, '“mm?”': 1, 'magic.”': 1, '“d’yeh': 1, 'anything?”': 1, '“they': 1, 'didn’': 1, 'gringotts': 5,
'wizards’': 1, 'wouldn’': 1, 'teh': 1, 'neither.”': 1, '“wizards': 1, 'banks?”': 1, '“just': 1, 'goblins.”': 1,
'“goblins?”': 1, '“yeah': 2, 'yeh’d': 1, '’cept': 1, 'business.”': 1, 'fetchin’': 1, '“got': 1, 'everythin’': 1,
'then.”': 1, '“how': 1, 'here?”': 1, '“flew,”': 1, '“flew?”': 1, 's’pposed': 1, 'yeh.”': 1}
```