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?

MY DOB IS 24/SEP/1998

the month is september(09), it is divided by 2 and rounded to 5. the 5th book is Harry Potter and the order of phonex

the date is 24 so I have extracted the 10 pages starting from the 24th page (file1)

. the year is 98, so i have extracted the 10 pages starting from 98th (page file2)

In [1]:
```
import re                       # importing the required packages
import string
from functools import reduce
from collections import Counter
```

In [2]:
```
(text):
e.findall(r'\w+', text.lower()) #  regular expression '\w+' takes whole t
rds                                     # returning the list of words
```

In [3]:
```
 open('file1.txt', 'rb') as file:                 # Opening the 'file1.t
 lines = file.readlines()                         # Reading all lines fr
 lines = [line.decode('utf-8') for line in lines]
 text = ''.join(lines)                            # Joining the list of
 words = map_words(text)                          # calling the 'map_wor
```

In [4]:
```
rd_counts_list = list(map(Counter, [words]))       # Using the map functi
tal_word_counts = reduce(lambda x, y: x + y, word_counts_list) # Using 'r

r word, count in total_word_counts.items():   # Iterateing through the to
  print(f'{word}: {count}')
```

```
of: 66
phoenix: 11
j: 11
k: 11
rowling: 11
barely: 2
an: 9
inch: 1
from: 13
dudley: 2
s: 32
when: 5
silver: 2
antlers: 1
caught: 1
it: 29
thing: 2
was: 45
thrown: 1
p: 11
g: 11
```

In [6]:
```
def map_words(text):
    return re.findall(r'\w+', text.lower())  # regular expression '\w+'
```

In [7]:
```
def is_non_english_word(word, spellchecker):
    return not word.isdigit() and word not in spellchecker   # Checking
```

In [8]:
```
Checker()                               # Initializing a SpellChecker ins
                                        # Initializing a dictionary to st

                                        # Opening 'file2.txt' in UTF-8 en
(                                       # Reading the entire content of t
s(text)                                 # calling the 'map_words' functio
s:                                      # Iterating through each word in
glish_word(word, spellchecker):         # Checking if the word is non-Eng
[word] = word_counts.get(word, 0) + 1 # If the word is not in 'word_co
ord_counts.items():                     # Iterating through the 'word_cou
{count}')
```

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 [5]:
```
import re
from functools import reduce
from spellchecker import SpellChecker
```

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 [6]:
```
def map_words(text):
    return re.findall(r'\w+', text.lower())  # regular expression '\w+'
```

In [7]:
```
def is_non_english_word(word, spellchecker):
    return not word.isdigit() and word not in spellchecker   # Checking
```

In [8]:
```
Checker()                                    # Initializing a SpellChecker ins
                                             # Initializing a dictionary to st

                                             # Opening 'file2.txt' in UTF-8 en
                                             # Reading the entire content of t
s(text)                                      # calling the 'map_words' functio

s:                                           # Iterating through each word in
glish_word(word, spellchecker):              # Checking if the word is non-Eng
    word_counts[word] = word_counts.get(word, 0) + 1 # If the word is not in 'word_co

ord_counts.items():                          # Iterating through the 'word_cou
{count}')
```

In [5]:
```
import re                                    # importing the required pack
from functools import reduce
from spellchecker import SpellChecker
```

```
p: 11
g: 11
j: 11
k: 11
rowling: 11
s: 40
irritably: 1
t: 12
heatedly: 1
kreacher: 1
c: 2
m: 5
mrs: 17
weasley: 21
tonks: 8
ll: 2
exasperatedly: 1
earsplitting: 1
yoooou: 1
mr: 4
greenland: 1
evanesco: 1
mundungus: 13
n: 1
sleepily: 1
gree: 1
balefully: 1
arry: 1
pology: 1
crookshanks: 2
absentmindedly: 1
incredulously: 1
d: 2
animagus: 1
wormtail: 1
bracingly: 1
butterbeer: 3
breadboard: 1
```

In [ ]:

In [ ]: