For this assignment, download the Harry Potter Books data from the following link (PDF is also attached):

https://ztcprep.com/library/story/Harry_Potter/Harry_Potter_(www.ztcprep.com).pdf

Extract Data:

Select the book which corresponds to your birth month. For birth month 8-12, divide by 2 and round up.

Once you selected the book, go to page number that corresponds to your birth date (1-31) and extract next 10 pages of the book to a text file (file1.txt).

Next, go to page number that corresponds to your birth year (last 2 digits). For year 2000 onwards, use 1 infront of the year number to find the page number (so year 2000 becomes 100, 2001 - 101 and so on). Extract next 10 pages into another text file (file2.txt).

Write Code to analyze data:

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?
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.

DOB : 08-08-1999

- Taking book number 4 as my book as 8 divide by 2 is 4

Question 1

In [3]:

```
pip install mrjob
```

```
Requirement already satisfied: mrjob in /Users/funnysmac/opt/anaconda3/lib/pyt
hon3.9/site-packages (0.7.4)
Requirement already satisfied: PyYAML>=3.10 in /Users/funnysmac/opt/anaconda3/
lib/python3.9/site-packages (from mrjob) (6.0)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:

```
%%file wordcount.py

from mrjob.job import MRJob
```

```
class wordcount(MRJob):

    def mapper(self, _, line):
        line = line.strip()
        words = line.split()
        for word in words:
            yield word, 1

    def reducer(self, word, counts):
        yield word, sum(counts)

if __name__ == '__main__':
    wordcount.run()
```

```
Overwriting wordcount.py
```

In [5]:

```
import wordcount
mr_job = wordcount.wordcount(args=["file1.txt"])
with mr_job.make_runner() as runner:
    runner.run()
    for key, value in mr_job.parse_output(runner.cat_output()):
        print(key, value)
```

```
No configs specified for inline runner
```

```
over 5
pain 1
painful. 1
panic 1
paralyzed 1
part 1
passageway. 1
passing; 1
path 1
pause 1
pause, 1
peered 1
people 1
perform 1
perform. 1
perhaps 1
person 1
pet 1
picture 2
plan 3
plan, 2
planning 2
plotting 2
pointed 1
police 1
police. 1
possible 1
powerful 1
present 1
pressed 2
problem 1
proceed, 1
promise 1
pronounced 1
protected 1
protection 1
protracted, 1
proved 1
put 2
puzzled 1
questioned 1
questioning, 1
questions 1
quickly! 1
quiet 1
quietly 2
quietly. 2
quite 1
raised 1
ran 1
rather 1
reaching 1
real. 1
realized 1
reasons 1
recall 1
reflection 1
regained 1
regretting 1
rejoined 1
```

```
relieved 1
remained 1
remember 2
remember. 1
remorse 1
requirement.” 1
returned 1
revolt 1
reward, 1
right 2
right?” 1
rising 1
rob 1
room 3
room, 5
room. 1
room; 1
rotting 1
roughly. 1
rug 2
rug, 1
rug. 1
run 1
running. 1
rush, 1
rushing 1
said 19
said. 1
sat 2
saw 1
say 2
say.” 1
scar 3
scar, 1
scar? 1
scrambled 1
scream. 1
screaming 1
screwed 1
second 9
seconds, 2
see 4
seem 2
seemed 2
seen 1
seizure. 1
sense,” 1
servant 3
servant, 1
servant,” 1
shadows 1
shaken, 1
shaking 1
shaped 1
she 4
short 1
short, 1
shudder 1
silent 1
silkily, 1
```

```
sitting 2
skin. 1
skinny 1
slight 1
slightly 1
slipped 1
slippery 1
slithering 1
sliver 1
small 2
small, 1
smaller 1
snagged 1
snake 5
snake, 2
snakes. 1
so 9
softly, 1
softly. 1
some 3
somebody 2
someone 3
something 1
sort 2
sound, 1
sounded 2
sounding 2
source 1
spasm 1
speak!” 1
speaker. 1
speaking 1
speech. 1
spidery 1
spies 1
spitting 1
spitting, 1
spoil 1
spoke 2
spoke, 1
spot, 1
sputtering 1
squeakily. 1
standing 1
stared 3
start. 1
started 1
stayed 1
steadier. 1
stick 5
still 5
still. 1
stinging. 1
stomach 1
stood 2
straight 1
strange 1
street 1
stroke 1
stronger, 2
```

```
substitute? 1
such 1
sudden 1
suddenly 2
suggestion 1
suitable 1
sulky 1
sullenness 1
supposed 1
surely 1
surprise? 1
surrounding 1
survive 1
sweat 1
sweat. 1
switched 1
swung 1
table. 1
tail 1
talk 1
talking 2
continuing 1
coolly. 1
corridor 1
corridor, 1
could 11
couldn’t 2
courage 2
cowardice. 1
crackling 1
crackling. 1
creep 1
criminals. 1
crossed 1
cruel 1
crumpled. 1
cube 1
cupped 1
curled 1
curtains 1
curving 1
cut 1
danger 1
dangerous 1
dark 2
darkened 1
darkness 1
days 2
dead 2
death 1
decided. 1
decision, 1
deeply 1
defiantly, 1
definitely 2
deny 1
desert 1
details 1
determined, 2
devotion 2
```

```
diamond-patterned 1
didn’t 3
didn’t. 1
died. 1
difference. 1
difficult, 1
dim 1
dimly 1
disappearance 1
disguise 1
do 6
do. 1
do? 1
dog. 1
don’t 5
done 4
door 1
door, 1
door. 1
down 1
drawing 1
dream 1
dreaming 1
drew 1
dry. 1
dust 1
e 9
edge 1
effective. 1
effectively 1
else 3
end 1
enough 2
entirely 1
escape 1
essential 1
even 2
ever 1
ever, 1
ever. 1
every 2
everything, 1
examined 1
expecting 1
explained 1
extent 1
extracted 1
eyes 3
eyes, 1
face 5
face. 2
facing 1
faint, 1
faithful 3
fall 1
fast 1
fear 1
feeding 1
Wormtail,” 2
Wormtail. 4
Wormtail.” 3
Wormtail’s 1
You 4
You’ve 1
Your 2
— 9
— 20
—” 10
’round 1
“A 1
“According 1
“Ah, 1
“All 2
“And 1
“But 2
“Certainly 1
“Do 1
“However, 1
“I 12
“If? 1
“If?” 1
“In-indeed, 1
“Indeed, 1
“Invite 1
“Is 1
“It 2
“Laying 1
“Liar,” 2
“Lord, 1
“My 4
“Nagini 1
“No! 1
“Nobody 1
“One 1
“R-really, 1
“That 1
“The 1
“We 1
“What’s 1
“Without 1
“Wormtail, 2
“Wormtail? 1
“You 5
“Your 2
“my 1
“that 1
“why 1
a 56
abandoning 1
about 3
abruptly 1
action, 1
added, 1
after 1
again, 2
again. 3
alarm 1
all! 1
all 3
allow 1
```

```
allowed 1
almost 1
alone 1
alone. 1
along 1
already 1
always 3
am 8
amused. 1
amusement 1
amusement. 1
an 8
ancient 1
and 55
another 2
any 3
anyone 1
anything 2
anywhere 1
approach 1
are 6
argument. 1
armchair 1
armchair; 1
around 1
around, 1
around.” 1
as 23
asked 1
asking 1
at 12
attempt 1
audible 1
aware 2
away 1
away, 1
awkward 1
awoken 2
awoken. 1
back 7
back, 1
balding 1
barely 1
be 18
became 1
because 1
beckoned 1
become 1
becoming 1
bed 1
bed, 1
bedroom 2
bedroom, 1
bedside 1
been 11
been? 1
before 6
before; 1
began 1
behind 1
```

```
being 1
believe 1
beneath 1
beside 1
beyond 1
black 1
blocking 1
body 1
bolt 1
bottle. 1
box 1
boy!” 1
boy 3
boy, 3
brains, 1
braver; 1
breath. 1
breathed 2
breathing 1
bright 1
brilliance 1
broken 1
brought 1
burning 1
but 9
by 5
called 2
calling 2
came 2
can 3
care. 1
case, 1
cast 1
caught 1
chair 5
chair. 1
chance 1
changed. 1
clatter. 1
clear. 1
clearer 1
closed 1
closely 1
closely. 1
closer 2
clumsy 1
code: 1
cold 13
cold, 1
come 4
coming 2
coming. 1
concentrated 1
concern 1
condition 1
confused. 1
this 6
thoroughly 1
though 8
though, 1
```

```
thought 3
thought. 1
threshold. 1
thrill 1
through 4
tightened 1
tightly 1
time 2
time, 1
tip 1
to 61
to. 1
told 1
told, 1
tonight, 1
too,” 1
too?” 1
took 2
touch 1
toward 1
trace 1
track 1
transfixed, 1
travesty 1
trembling. 1
triangular 1
trickling 1
tried 3
true,” 1
true. 1
truth 1
trying 4
tumbling 1
turn 2
turned 2
twelve 1
two 3
ugly 1
under 2
understand 1
undulating 1
unfortunately, 1
unless 1
unnoticed 1
untidy 1
up 2
up, 2
upon 2
us 1
use 4
useful 2
useful, 1
useless. 1
using 1
vanished 1
very 4
village 1
visited 1
vivid 1
voice 20
```

```
voice, 2
voice. 6
volunteer 1
waited 1
walked 1
walking 5
walls. 1
wand. 1
want 2
wanted 1
war. 1
wardrobe, 1
was 50
was, 1
watched 1
water 1
watery 1
wavered, 1
wayside 1
we 3
wearisome 1
well 2
were 7
what 7
when 6
where 4
which 3
while 1
whimper. 1
whisper 1
whispered 1
white-hot 1
who 4
whoever 1
whose 1
why 2
wide 1
wide, 1
wife 1
wife,” 1
will 13
window. 1
wire 1
wish 3
witch 1
witches 1
with 16
without 5
wizard 2
wizard, 1
wizard,” 2
wizard.” 1
woke 1
woman. 1
wonder 1
word 1
words 2
would 13
wrath 1
www.ztcprep.com 16
```

```
years. 1
yes,” 1
yet 1
you 30
you, 8
you,” 1
you. 3
you? 1
you?” 2
you’re 2
your 4
| 9
task 2
telephone 1
tell 1
tell, 1
terrified 1
terror 1
than 7
that 26
that, 1
the 134
their 1
them 1
them. 1
then 5
then, 1
then?” 1
there 4
they 2
thick 1
thing 2
think 2
thirteen 1
it 18
it, 3
it. 1
it? 1
its 4
journey 1
just 2
keep 1
kill 4
killed 2
killed, 1
kind 1
knew 3
know 4
knows 3
knows. 1
lamp 2
laugh 1
laugh, 1
lay 1
lay, 1
least 1
leave 2
legs 2
let 1
level 1
```

```
lie 2
lifted 1
light 2
light, 1
lightning, 1
lightning-bolt 1
like 4
like, 1
limped 1
listened 1
listening 2
lit 2
little 3
long, 2
long. 1
look 1
look, 1
looked 3
lost 1
loudly 1
loyalty 1
made 3
madman. 1
make 1
makes 1
making 1
man 9
man, 3
man. 1
man; 1
man’s 1
manners, 1
manners?” 1
many 1
master 2
me!” 1
me 7
me, 4
me. 2
me?” 2
mean 1
means 2
memory 1
memory? 1
men 1
menace 1
merely 1
merest 1
met 1
might 1
might. 1
miles 1
milk 1
mine, 1
miraculously, 1
mirror 1
mirthless 1
misty 1
mixture 1
modified 1
```

```
moment 1
months 1
more! 1
more 14
I’ll 1
I’m 1
I’ve 1
If 2
In 1
Inside 1
It 7
J.K. 9
Jorkins.” 2
Jorkins’s 1
Lord! 1
Lord 5
Lord, 4
Lord. 1
Lord.” 1
Lord? 1
Lord?” 1
Lordship 2
Magic 1
Memory 1
Ministry 3
Muggle 1
Muggle, 1
Muggle,” 2
Muggle?” 1
My 8
Nagini, 1
Nagini. 1
Nagini?” 1
Now, 1
Out 1
P 9
Peter, 1
Potter 13
Potter, 2
Potter?” 1
Rowling 9
SCAR 1
She 1
Silence!” 1
Slowly, 1
Something 1
THE 1
The 10
Then 1
There 7
This 1
Though 1
Turn 1
Two 1
Voldemort 2
Voldemort, 1
Voldemort. 1
Voldemort’s 2
Well, 1
What 2
```

```
Where 1
Who 1
Without 1
Wizards 1
Wormtail! 1
Wormtail 9
Wormtail, 10
feel 2
feet 1
fell 2
felt 3
fetch 1
few 4
filtering 1
find, 1
fingers 2
fire 2
fire, 1
firelight, 1
firmer 1
fit 2
flames. 1
flash 1
flat 1
flinch 1
floor 1
floor, 2
floor. 1
flung 1
focus, 1
follow 1
followed 1
followers 1
following 1
footsteps, 1
for 15
forcing 1
forehead 1
forehead, 1
formed 1
forward 1
found 3
fourteen 1
fright. 1
from 7
frowning, 1
frozen 1
fulfill 1
full 1
fuss; 1
g 9
gap. 1
gave 1
gigantic 1
give 1
glasses, 1
go 3
go. 1
going 2
gone 3
```

```
good 1
graying 1
green 2
grip 1
gripping 1
ground. 1
growing 1
had 33
hair, 1
hair. 1
hand 3
hand, 1
hands 3
hands, 2
hands; 1
hard 1
hard, 1
has 4
have 16
have. 1
having 1
he 31
he, 1
head 3
head. 1
health 1
hear 3
heard 6
hearth 3
her 4
her, 2
her. 1
here 2
here, 1
here. 1
hide 1
high 1
him! 1
him 7
him, 3
him. 1
himself 2
himself, 1
himself. 1
his 48
hiss, 1
hiss. 1
hissed 2
hissing 2
hit 1
hoarse, 1
hold 3
holidays. 1
honor 1
horrible 1
horror, 1
hot-water 1
hours? 1
house 1
how 1
```

```
hundred 1
ice 1
idea, 1
idea. 1
if 9
impossible 1
impossible. 1
in 23
incoherently, 1
incredibly, 1
information 2
inns. 1
inside 3
inside, 1
inspiration, 1
insult 1
interest 1
interesting 1
into 7
invaluable. 1
is 16
... 27
...” 8
10 1
11 1
12 1
13 1
14 1
15 1
16 1
17 1
18 1
?” 1
A 4
All 2
And 6
Another 1
As 2
Bertha 4
Bryce 1
But 3
By 1
Charms 1
Come, 1
Could 1
Do 1
English 1
Fire 9
For 2
Frank 21
Frank, 3
Frank’s 1
Goblet 9
Harry 25
Harry, 1
He 16
His 1
Hogwarts 1
Horrified, 1
How 1
```

```
However 1
I 44
more, 2
most 1
mouth 2
move. 1
movement 1
much 4
much, 1
murder 3
murder, 1
murders 1
must 3
muttered 1
my 9
myself 1
myself, 1
name 1
nearer 1
need 3
needed 1
neither 1
nerve. 1
never 5
news 1
news, 1
nicknamed 1
no 6
nobody 1
noises 2
none 1
normal, 1
nose, 1
not 15
not? 1
note 1
nothing 6
now 4
now, 1
now. 4
nursing 1
of 48
old 4
on 20
on. 1
once 2
once. 1
one 5
only 3
open. 1
opened 3
or 3
orange 1
other 2
other. 1
our 3
out 7
outside 2
```

Question 2

In [6]:

```
pip install pyspellchecker
```

```
Requirement already satisfied: pyspellchecker in /Users/funnysmac/opt/anaconda
3/lib/python3.9/site-packages (0.7.1)
Note: you may need to restart the kernel to use updated packages.
```

In [7]:

```
from spellchecker import SpellChecker
spell = SpellChecker()
```

In [16]:

```
from nltk.tokenize import word_tokenize
import string
import codecs
import re
with codecs.open("english3.txt", 'r', encoding='utf-8',errors='ignore') as f1:
    f1 = f1.read()
    f1v = word_tokenize(f1)
```

In [8]:

```
from nltk.tokenize import word_tokenize
def tokenize(word):
    tokens = word_tokenize(word)
    tokens = [w.lower() for w in tokens]
    return set(tokens)
```

In [39]:

```
def mapper(txt):
    for word in tokenize(txt):
        yield(word, 1)

def word_count(txt):
    x=[]
    for word, count in mapper(txt):
        if word not in f1v:
            print(word)
            x.append(word)
    return x
```

In [40]:

```
file2 = open("file2.txt", "r")
file2 = file2.read()
li=word_count(file2)
```

```
less-than-
205
right-hand
202
out-of-bounds
(
www.ztcprep.com
slytherin
er-my-knee
208
hagrid
triwizard
:
move.
thirty-
quidditch
mr.
champion.
;
...
couldn
”
discontinued.
wasn
year.
underage
.
209
j.k.
|
“
)
—
whole-hearted
’
it.
short-listed
206
isn
wouldn
—
hogwarts
yo-yos
ll
house-elves
beauxbatons
?
207
weasley
albus
204
durmstrang
ever-bashing
horror-struck
inter-house
203
201
hogsmeade
hmph
names.
```

```
rowling
money.
,
ve
moody.
hundred.
house-elf
down.
!
mad-eye
```

In [42]:

```
d ={}
for i in range(len(li)-1):
    x=li[i]
    c=0
    for j in range(i,len(li)):
        if li[j]==li[i]:
            c=c+1
    count=dict({x:c})
    if x not in d.keys():
        d.update(count)
print (d)
```

```
{'less-than-': 1, '205': 1, 'right-hand': 1, '202': 1, 'out-of-bounds': 1,
'(': 1, 'www.ztcprep.com': 1, 'slytherin': 1, 'er-my-knee': 1, '208': 1, 'hagr
id': 1, 'triwizard': 1, ':': 1, 'move.': 1, 'thirty-': 1, 'quidditch': 1, 'm
r.': 1, 'champion.': 1, ';': 1, '...': 1, 'couldn': 1, '”': 1, 'discontinue
d.': 1, 'wasn': 1, 'year.': 1, 'underage': 1, '.': 1, '209': 1, 'j.k.': 1,
'|': 1, '“': 1, ')': 1, '—': 1, 'whole-hearted': 1, '’': 1, 'it.': 1, 'short-l
isted': 1, '206': 1, 'isn': 1, 'wouldn': 1, '–': 1, 'hogwarts': 1, 'yo-yos':
1, 'll': 1, 'house-elves': 1, 'beauxbatons': 1, '?': 1, '207': 1, 'weasley':
1, 'albus': 1, '204': 1, 'durmstrang': 1, 'ever-bashing': 1, 'horror-struck':
1, 'inter-house': 1, '203': 1, '201': 1, 'hogsmeade': 1, 'hmph': 1, 'names.':
1, 'rowling': 1, 'money.': 1, ',': 1, 've': 1, 'moody.': 1, 'hundred.': 1, 'ho
use-elf': 1, 'down.': 1, '!': 1}
```

In [ ]: