# Harry Potter MapReduce Word Count

```
Input:
DOB: 19-05-2003
Book: Book 5
Pages used for file1 and file2.
```

```
# DOB: 19-05-2003

import sys

for line in sys.stdin:
    words = line.strip().split()
    for word in words:
        print(f"{word}\t1")
```

```
import sys

current_word = None
current_count = 0

for line in sys.stdin:
    word, count = line.strip().split("\t")
    count = int(count)

    if current_word == word:
        current_count += count
    else:
        if current_word:
            print(f"{current_word}\t{current_count}")
        current_word = word
        current_count = count

if current_word:
    print(f"{current_word}\t{current_count}")
```

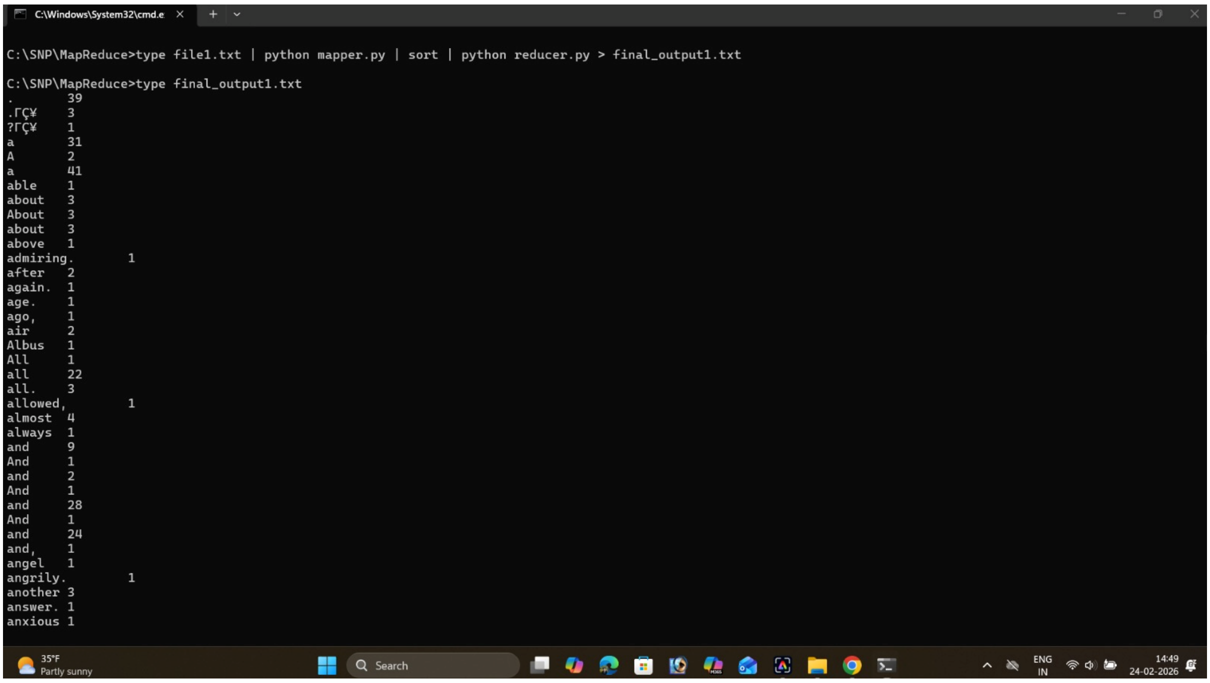
n this task I used the MapReduce approach to count how many times each word appears in the first text file.

The mapper reads the file line by line, removes punctuation, converts the words to lowercase, and outputs each word with the value 1.

After that, the reducer takes all the values for the same word and adds them together to get the final count.

This process is useful for large datasets because the work can be distributed across multiple systems, but here I tested it on a single node Hadoop setup.

The final output shows each word and how many times it is repeated in the file.



```
C:\SNP\MapReduce>type file1.txt | python mapper.py | sort | python reducer.py > final_output1.txt

C:\SNP\MapReduce>type final_output1.txt
.       39
.ΓÇ¥    3
?ΓÇ¥    1
a       31
A       2
a       41
able    1
about   3
About   3
about   3
above   1
admiring.       1
after   2
again.  1
age.    1
ago,    1
air     2
Albus   1
All     1
all     22
all.    3
allowed,        1
almost  4
always  1
and     9
And     1
and     2
And     1
and     28
And     1
and     24
and,    1
angel   1
angrily.        1
another 3
answer. 1
anxious 1
```

# Question:2

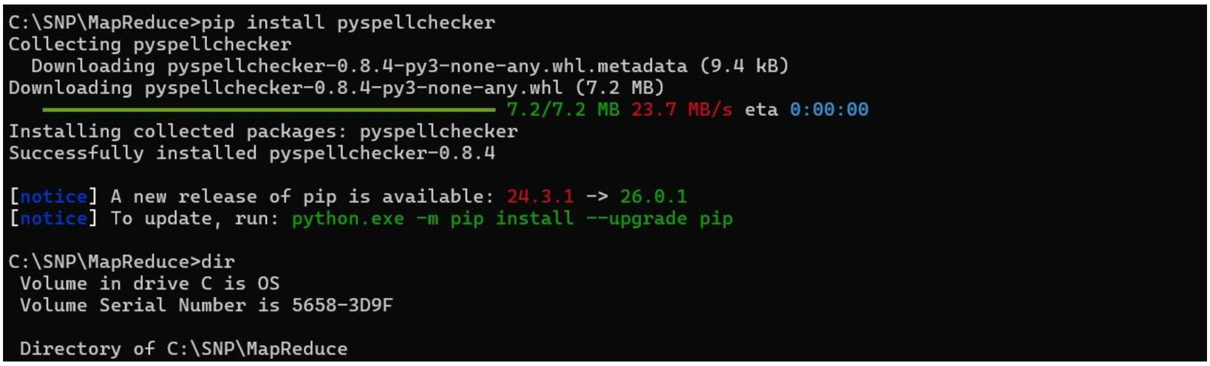
```
# DOB: 19-05-2003

import sys
import re
from spellchecker import SpellChecker

spell = SpellChecker()

for line in sys.stdin:
    words = re.findall(r'\b[a-zA-Z]+\b', line.lower())

    for word in words:
        if word not in spell:   # non-English words
            print(f"{word}\t1")
```

```
import sys

current_word = None
current_count = 0

for line in sys.stdin:
    word, count = line.strip().split("\t")
    count = int(count)

    if current_word == word:
        current_count += count
    else:
        if current_word:
            print(f"{current_word}\t{current_count}")
        current_word = word
        current_count = count

if current_word:
    print(f"{current_word}\t{current_count}")
```

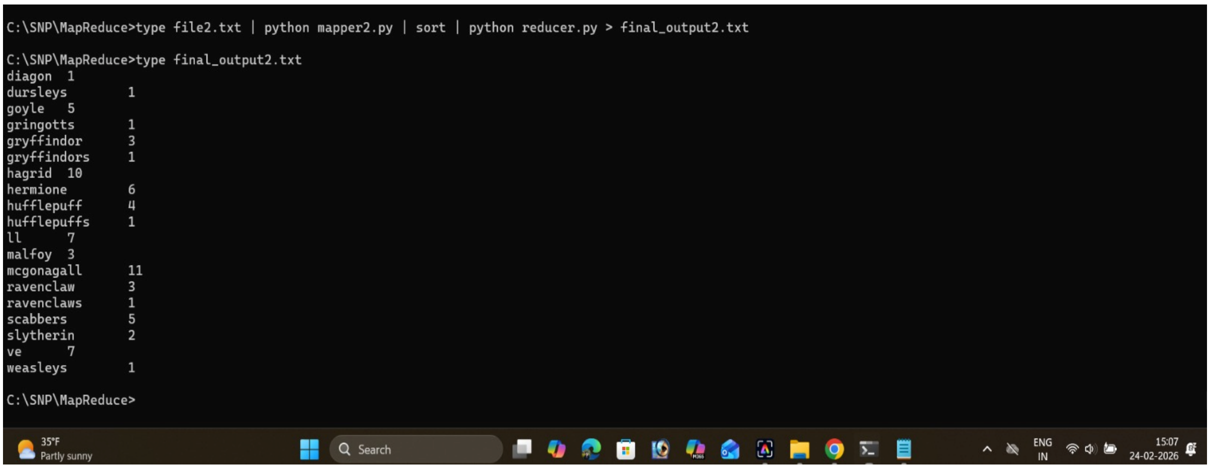
In the second task the goal was to find non-English words such as names, places, and spell words from the second text file.
For this I used the pyspellchecker library to compare each word with a standard English dictionary.
If the word was not found in the dictionary, it was treated as a non-English word.

The mapper outputs those filtered words with the value 1, and the reducer adds the counts to show how many times each non-English word appears.

This method helped to identify special words like character names and locations that are not part of normal English vocabulary.



```
C:\SNP\MapReduce>pip install pyspellchecker
Collecting pyspellchecker
  Downloading pyspellchecker-0.8.4-py3-none-any.whl.metadata (9.4 kB)
Downloading pyspellchecker-0.8.4-py3-none-any.whl (7.2 MB)
   ---------------------------------------- 7.2/7.2 MB 23.7 MB/s eta 0:00:00
Installing collected packages: pyspellchecker
Successfully installed pyspellchecker-0.8.4

[notice] A new release of pip is available: 24.3.1 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\SNP\MapReduce>dir
 Volume in drive C is OS
 Volume Serial Number is 5658-3D9F

 Directory of C:\SNP\MapReduce
```



```
C:\SNP\MapReduce>type file2.txt | python mapper2.py | sort | python reducer.py > final_output2.txt

C:\SNP\MapReduce>type final_output2.txt
diagon  1
dursleys        1
goyle   5
gringotts       1
gryffindor      3
gryffindors     1
hagrid  10
hermione        6
hufflepuff      4
hufflepuffs     1
ll      7
malfoy  3
mcgonagall      11
ravenclaw       3
ravenclaws      1
scabbers        5
slytherin       2
ve      7
weasleys        1

C:\SNP\MapReduce>
```

GitHub Repo Link:
https://github.com/sandeepnaidupenta/harry-potter-mapreduce