CS 5665, HW 4
Karun Joseph, A02240287

All code, files reside here:
https://github.com/karunmj/usu-coursework/tree/master/cs5660datasc/hw/hw4

The mapper and reducer functions for respective tasks are included as appendix.

Task 1: Word count

1. Ordinary words
   a) Complete count of each word
   Refer to outputwp.txt file in GitHub repo

   b) Word that appears the most
   'the' appears the most with 34254 occurrences, followed by 'and' with 21389 occurrences.

2. Palindrome words
   a) Complete count of each word
   Refer to outputwp_pd.txt file in GitHub repo

   b) Word that appears the most
   'a' appears the most (10408). If 'a' cannot be considered as a word (also 'i'), then 'did' (1427) appears the most.

Task 2: Election fraud

1. Party that won the election in 2008
   '3' party has won the election in 2008 with 12071votes

	Party 1	Party 2	Party 3
Number of votes	9408	10112	12071

2. County that was most monolithic in the manner they voted in 2006
   '277' county was the most monolithic with 52.5% voting for party '3'

3. Counties where voter fraud has occurred in 2008
   'x', 'y' and 'z' are the counties in which voter fraud has occurred (I was able to write mapper and reducer functions mostly!)

4. Number of voters who changed the party they voted from 2006 to 2008
   6297 voters have changed the party they voted from 2006 to 2008. The most changes were from party 'x' to 'y'.

Appendix: Mapper and Reducer functions

Task 1: Word count

1. Ordinary words
   a) Complete count of each word
   b) Word that appears the most

CS 5665, HW 4
Karun Joseph, A02240287

Mapper

```
#!/usr/bin/env python

import sys

for line in sys.stdin:
    line = line.strip()
    for word in line.lower().split():
        print '%s\t%s' % (word, "1")
```

Reducer

```
#!/usr/bin/env python

import sys

word2count = {}

for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t', 1)
    try:
        count = int(count)
    except ValueError:
        continue
    try:
        word2count[word] = word2count[word]+count
    except:
        word2count[word] = count

for word in word2count.keys():
    print '%s\t%s' % (word, word2count[word])
```

2. Palindrome words
    a) Complete count of each word
    b) Word that appears the most

    Mapper

    ```
    #!/usr/bin/env python

    import sys

    for line in sys.stdin:
        line = line.strip()
        for word in line.lower().split():
            if str(word) == str(word)[::-1]:
                print '%s\t%s' % (word, "1")
    ```

Reducer

```
#!/usr/bin/env python

import sys

word2count = {}

for line in sys.stdin:
    line = line.strip()
    word, count = line.split('\t', 1)
    try:
        count = int(count)
    except ValueError:
        continue
    try:
        word2count[word] = word2count[word]+count
    except:
        word2count[word] = count

for word in word2count.keys():
    print '%s\t%s' % (word, word2count[word])
```

Task 2: Election fraud

1. Party that won the election in 2008

   Mapper

   ```
   #!/usr/bin/env python

   import sys

   for record in sys.stdin:
       record = record.strip()
       print '%s\t%s' % (record.split('\t')[2], "1")
   ```

   Reducer

   ```
   #!/usr/bin/env python

   import sys

   vote2count = {}

   for line in sys.stdin:
       line = line.strip()
       vote, count = line.split('\t', 1)
       try:
           count = int(count)
       except ValueError:
   ```

```
        continue
    try:
        vote2count[vote] = vote2count[vote]+count
    except:
        vote2count[vote] = count

for vote in vote2count.keys():
    print '%s\t%s' % (vote, vote2count[vote])
```

2. County that was most monolithic in the manner they voted in 2006

Mapper

```
#!/usr/bin/env python

import sys

for record in sys.stdin:
    record = record.strip()
    print '%s\t%s\t%s' % (record.split('\t')[1], record.split('\t')[2], "1")
```

Reducer

```
#!/usr/bin/env python

import sys

countybyparty = {}
countyperc = {}

for line in sys.stdin:
    line = line.strip()
    county, partyid, count = line.split('\t')

    try:
        county = int(county)
    except ValueError:
        continue

    try:
        partyid = int(partyid)
    except ValueError:
        continue

    try:
        count = int(count)
    except ValueError:
        continue
    try:
        countybyparty[county][partyid] = countybyparty[county][partyid] + count
    except:
        try:
            countybyparty[county][partyid] = count
        except:
            countybyparty[county] = {}

for county in countybyparty:
    #print '%s\t%s' % (county, countybyparty[county])
    a = []
    for party in countybyparty[county]:
        a.append(countybyparty[county][party])
    print '%s\t%s' % (county, float(max(a))/sum(a))
```

3. Counties where voter fraud has occurred in 2008

<u>Mapper</u>

```
#!/usr/bin/env python

import sys

for record in sys.stdin:
    record = record.strip()
    print '%s\t%s\t%s\t%s' % (record.split('\t')[1], record.split('\t')[2], record.split('\t')[5],
"1")
```

<u>Reducer</u>

```
#!/usr/bin/env python

import sys

countybyyear2006 = {}
countybyyear2008 = {}
#countyperc = {}

for line in sys.stdin:
    line = line.strip()
    county, partyid2006, partyid2008, count = line.split('\t')

    try:
        county = int(county)
    except ValueError:
        continue

    try:
        partyid2006 = int(partyid2006)
    except ValueError:
        continue

    try:
        partyid2008 = int(partyid2008)
    except ValueError:
        continue

    try:
        count = int(count)
    except ValueError:
        continue

    try:
        countybyyear2006[county][partyid2006] =
countybyyear2006[county][partyid2006] + count
    except:
        try:
            countybyyear2006[county][partyid2006] = count
        except:
            countybyyear2006[county] = {}

    try:
        countybyyear2008[county][partyid2008] =
countybyyear2008[county][partyid2008] + count
    except:
        try:
            countybyyear2008[county][partyid2008] = count
        except:
            countybyyear2008[county] = {}

for county2006, county2008 in zip(countybyyear2006, countybyyear2008):
    print '%s\t%s\t%s\t%s' % (county2006, countybyyear2006[county2006], county2008,
countybyyear2008[county2008])
    #print '%s\t%s' % (county2006, float(countybyyear2008[k])/countybyyear2006 for k in
countybyyear2006.viewkeys() & countybyyear2008.viewkeys())
```

4. Number of voters who changed the party they voted from 2006 to 2008

   Mapper

```
#!/usr/bin/env python

import sys

for record in sys.stdin:
    record = record.strip()
```

```
   print '%s\t%s\t%s' % (record.split('\t')[0], record.split('\t')[2], record.split('\t')[5])
```

Reducer

```
#!/usr/bin/env python

import sys

votechange = {}

for line in sys.stdin:
line = line.strip()
voterid, party2006, party2008 = line.split('\t')

try:
        voterid = int(voterid)
except ValueError:
        continue

try:
         party2006 = int(party2006)
except ValueError:
        continue

try:
        party2008 = int(party2008)
except ValueError:
        continue

try:
        if party2006!=party2008:
                votechange[voterid] = 1
        if part2006==party2008:
                votechange[voterid] = 0
except:
        pass


for voter in votechange:
        print '%s\t%s' % (voter, votechange[voter])
```