

APL Assignment 9

Name: Aishwarya Bhavsar

Student ID: 029371509

Source Code:

```
#include <iostream>

#include <fstream>

#include<iomanip>

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

//Created and used aggregated classes (Word and WordList)

class Word          //Class Word for storing word & its count.
{
    // Operator Overloading.

public :

    int wcount;

    string wcounts;

    Word (string data)
    {
        wcounts = data;

        wcount = 1;

    }

    bool operator < (Word const &a1)
    {
```

```

        return wcounts < a1.wcounts;
    }

    friend ostream & operator << (ostream & out, const Word & a)
    {
        out << left;

        out << setw (10) << a.wcounts << a.wcount;

        return out;
    }
};

class WordList                //Sequence Of Container.
{
    public:

    vector<string> vec1;        //Used a vector sequence container from the STL to store objects
    string words;

    void asc_sort()
    {
        sort(vec1.begin(), vec1.end());    //Sort algorithm applied - ascending order
    }

    void CountTheWords ()
    {
        int wordCount = 1;

        int serial_num = 0;

        words = vec1[0];

        int vsize = vec1.size();

        if (vsize == 0)

```

```
{  
    cout << "File is empty, no words present in the file" << endl;  
}
```

```
for(int i=1; i<vsize; i++)  
{  
    if(words!=vec1[i] && serial_num<=vec1.size())  
    {  
        serial_num++;  
        cout << serial_num << ". " << words << " " << wordCount << endl;  
        wordCount=1;  
        cout << "\n";  
        words=vec1[i];  
    }  
    else  
    {  
        wordCount++;  
    }  
}
```

```
int sn = serial_num+1;  
cout << sn << ". " << words << " " << wordCount << endl;  
cout << "\n";  
}
```

```
};
```

```
int main()
```

```

{

ifstream infile("file.txt");

WordList obj;
obj.vec1;
obj.words;
int wcount = 43;
cout << "D:\\> filescan file.txt";
cout << "\n\n";
cout << "File file1.txt contains " << wcount << " words.";
cout << "\n\n";
while(infile >> obj.words)
{
    obj.vec1.push_back(obj.words);

}

obj.asc_sort();
obj.CountTheWords();

}

```

File.txt

to be or not to be that is the question

whether tis nobler to er er ah gall nabit

and so on to not not not be is that a question

or is it slung arrow being a bare bodkin who

delay the law or be be be be be be be be be

love pangs but sleep of merit be or not to be that

the worthy takes but a sleep of dreams undiscovers

the country from who no borrower lends

Output:

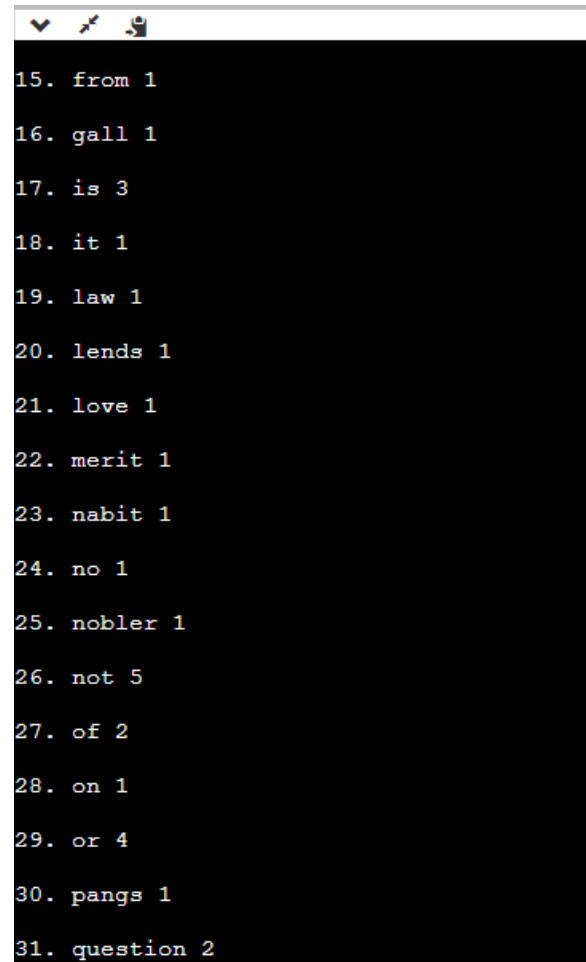
```

D:\> filescan file.txt

File file1.txt contains 43 words.

1. a 3
2. ah 1
3. and 1
4. arrow 1
5. bare 1
6. be 15
7. being 1
8. bodkin 1
9. borrower 1
10. but 2
11. country 1
12. delay 1
13. dreams 1
14. er 2
15. from 1

```



```
15. from 1
16. gall 1
17. is 3
18. it 1
19. law 1
20. lends 1
21. love 1
22. merit 1
23. nabit 1
24. no 1
25. nobler 1
26. not 5
27. of 2
28. on 1
29. or 4
30. pangs 1
31. question 2
```

```
29. or 4
30. pangs 1
31. question 2
32. sleep 2
33. slung 1
34. so 1
35. takes 1
36. that 3
37. the 4
38. tis 1
39. to 5
40. undiscovers 1
41. whether 1
42. who 2
43. worthy 1

...Program finished with exit code 0
Press ENTER to exit console.
```