In this lab we will learn how to search and sort linked lists. First, create a linked list to hold a char *. Your item will look as follows.

```
class Item
{
        public:
                Item *Next;
                char * Value;
                int Count; // Reference count, init to 1.
                Item( );  // set both pointers to NULL
                ~Item( );  // delete the value
                Item(char * NewString); // copies Value to NewString and sets Count to 1
                bool IsEqual(char * Data);   // returns true if strcmp(Data,Value)= = 0
                void AddRef(void); // adds 1 to Count;
};
```

Next, create a header class for the linked list.

```
class Header
{
        private:
                Item * Head,*Tail,*Current;
                int Count;
        public:
                Header( ); // Set Head, Tail, and Current to NULL, Count to zero;
                ~Header( ); // Destroy Linked List
                void AddRef(char * Data); // add 1 to ref count if we've seen the string
                                          // otherwise create string with ref count of 1.
                Item * First(void); // return Head and set Current to Head
                Item * Next(void); // Sets Current to Current->Next and returns Current
                                   // If Current is NULL, returns NULL
}
```

In the main program, open a file named "Words.txt" and read each word from the file. There is one word per line and there are no spaces in the file. Read each word, one at a time. The first time you encounter a word, the Header class must create a new Item to hold it. All subsequent encounters of the same word will increment the count by 1. At the end, list the words (in the order you encountered them) and print each word and its count on a separate line in the following format.

```
the---76
one---45
grade---45
ore---456
rocket---2
```

Print your output and turn it in along with a printout of your code.