The core of the data is stored in an array, songWords

```
songWords is an array of type number, size of 100

stuct number
      word: type string, holds the word
      value: type int, value of word counter
      <: overloaded operator for sorting structure

string promptForFile();
```

Prompts the user for the name of the file

do-while loop utilizing temp string and cin to grab user input

```
int readThisFile(number* songWords, string name);
```

Reads words from file and appends into array

Opens file and utilizes stringstream to get individual words

Cleanse words using regular expression to strip punctuation

Uses transform to ensure all words are lowercase

Check if word already exists in array

If yes, increments value and breaks out of loop

If no, adds value to last element and breaks out of loop

Program returns longest word in the file

```
int printMap(number* songWords, int longest);
```

Prints the array

Sorted array is passed to this function (sorting is done in main with overloaded operator for structure and C++ sorting algorithm)

Prints out sorted array (omits blanks in array)

Returns the most frequent word count

```
int countArray(number* songWords);
```

Counts the number of words

For loop to loop through array and count the number of total words stored, omits blanks

Returns int with the number of words

```
void writeToFile(number* songWords, int longest, int most, int count);
```

Writes the results to a file

Opens a file to prepare for writing

Writes information, including loop to write results of Array

Exits program