Anna Dubé

00091750

adube@my.athens.edu

Assignment 06

Assignment: use a file to create a dictionary. Every word that appears in the file should be a key word in the dictionary.

Test ID:			The file exists
	Preconditions	:	The file is stored in the correct folder with the correct name
	Input Data	:	file name
	Expected Result	:	the files will be opened and used by the program.
	Postconditions	:	The program continues
Test ID:			The file does not exist
	Preconditions	:	The file is not stored in the right folder or the file name is incorrect.
	Input Data	:	file name
	Expected Result	:	the file fails to open
	Postconditions	:	The program ends
Test ID:			the list is created using the file contents
	Preconditions	:	the file was opened
	Input Data	:	none
	Expected Result	:	the contents will be stored in the list
	Postconditions	:	the program continues
Test ID:			the queue is created
	Preconditions	:	the list was created
	Input Data	:	none
	Expected Result	:	the queue contains the first seven words in the list.
	Postconditions	:	the program continues

Test ID:		the queue is empty
	Preconditions :	the list was not created correctly or it was not read correctly to create the queue.
	Input Data :	none
	Expected Result :	there will be no data to work with
	Postconditions :	the program will continue but nothing will be displayed
Test ID:		The data is stored in the map
	Preconditions :	the queue was created
	Input Data :	none
	Expected Result :	the data is stored correctly
	Postconditions :	the program continues
Test ID:		The map is empty
	Preconditions :	the queue was created, but it is empty, or the data was not succssfully read and stored in the map.
	Input Data :	none
	Expected Result :	the data is not stored in the map
	Postconditions :	the program continues
Test ID:		the data is displayed incorrectly
	Preconditions :	data may not have been added to and removed from the queue correctly.
	Input Data :	none
	Expected Result :	not all of the keywords will be displayed
	Postconditions :	the program ends

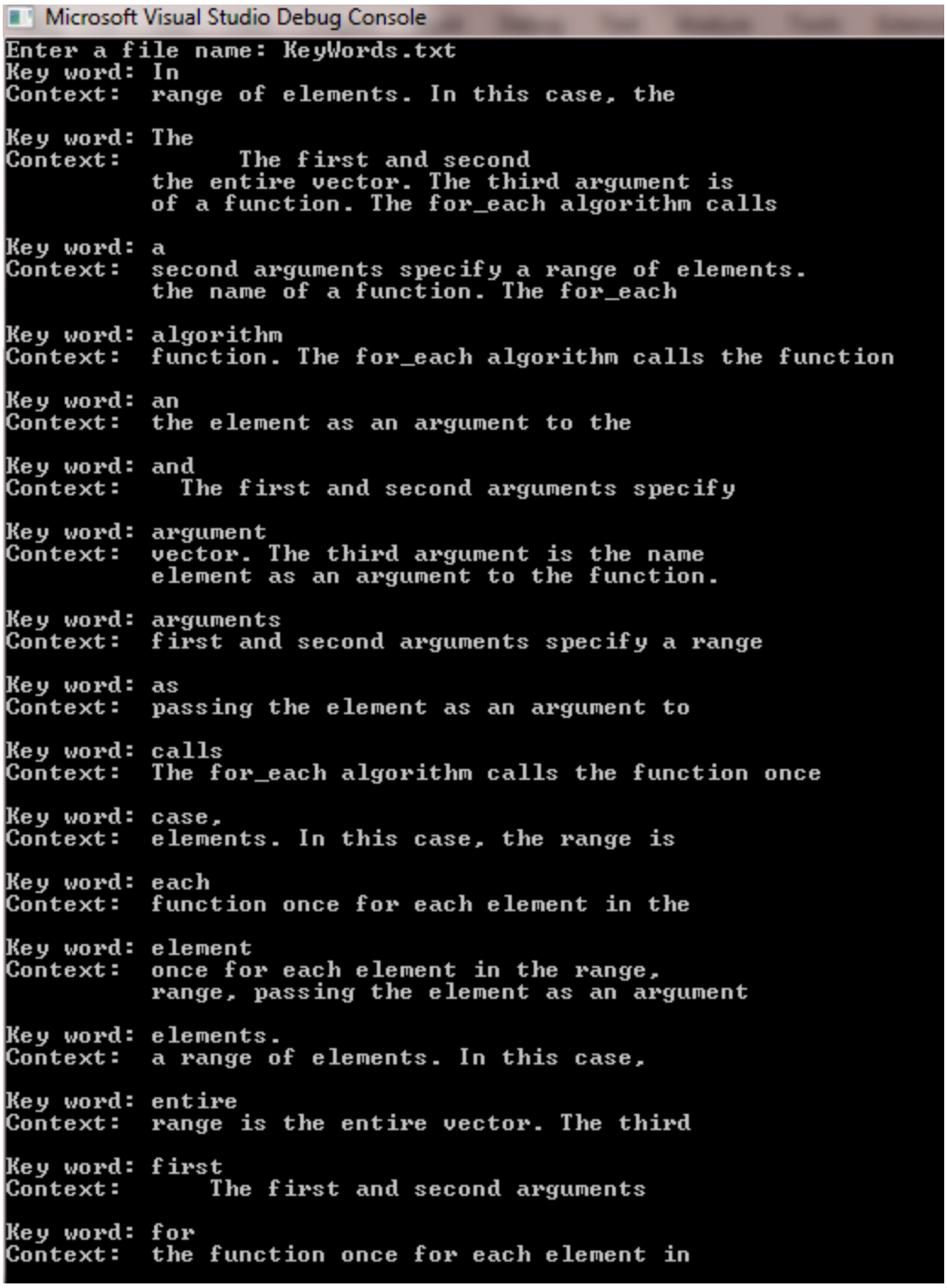
```
#include <iostream>
#include <list>
#include <fstream>
#include <map>
#include <queue>
using namespace std;

void mapDictionary();
list <string> words;
list<string>::iterator position;

int main()
{
	fstream wordFile;
	string name;
	cout << "Enter a file name: ";
	cin >> name;
	wordFile.open(name);

	if (wordFile.fail()) {
		cout << "Error\n";
```

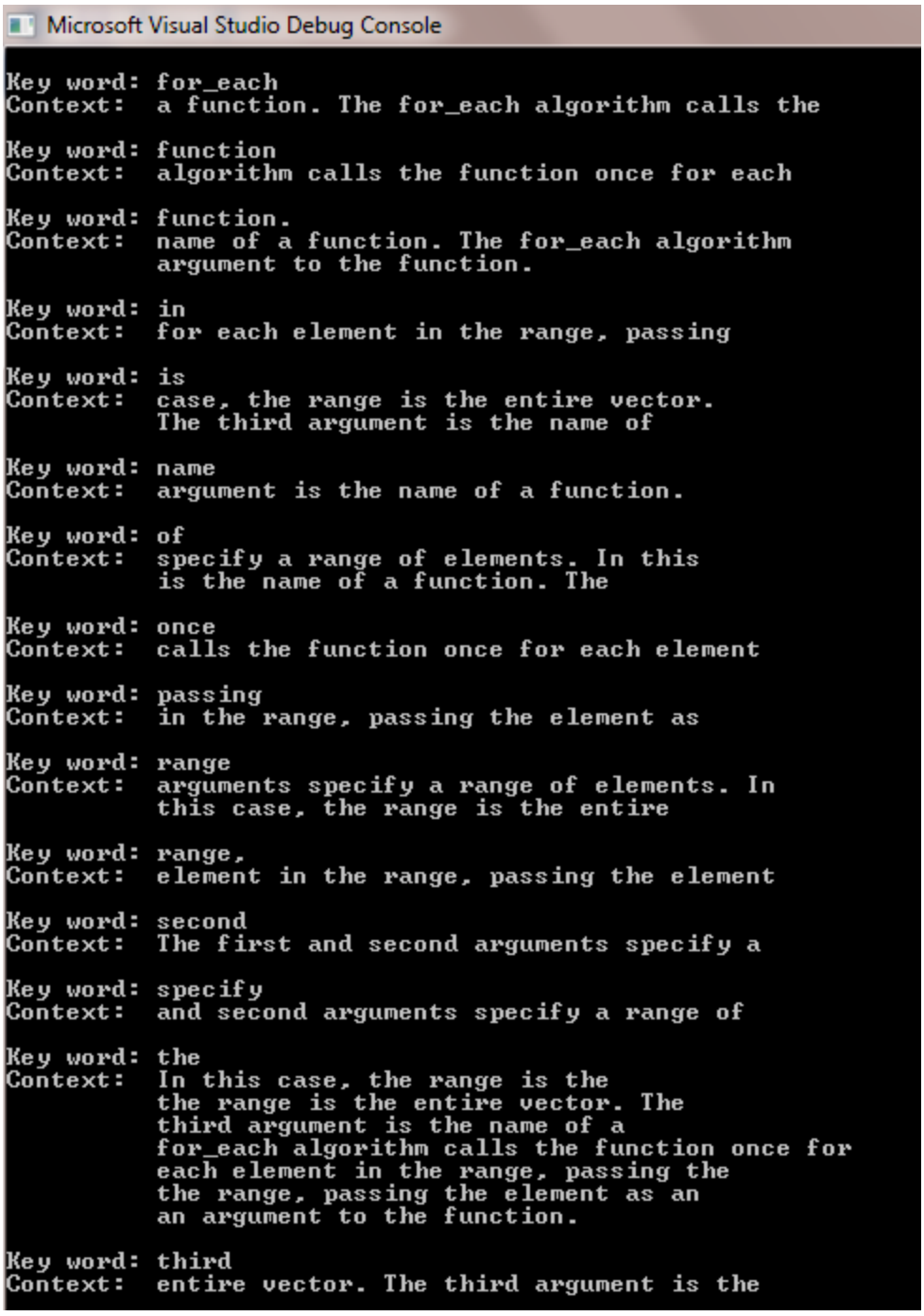
```
	}
	else {
		while (!wordFile.eof()) {
			string wordsInFile;
			wordFile >> wordsInFile;
			words.push_back(wordsInFile);
		}
		for (int x = 0; x < 3; x++) {
			words.push_back(" ");
			words.push_front(" ");
		}
		wordFile.close();
		mapDictionary();
	}
	return 0;
}

void mapDictionary() {
	multimap<string, string>data;
	queue<string>q;
	string temp = "";
	multimap<string, string>::iterator pos;
	multimap<string, string>::iterator iter;

	int x = 0;
	for (position = words.begin(); position != words.end(); ++position) {
		++x;
		if (x == 8) {
			break;
		}
		q.push(*position);
	}

	position = words.begin();
	for (; position != words.end(); ++position) {
		temp = " ";
		if (*position != " ") {
			for (int x = 0; x < 3; ++x) {
				--position;
			}
			for (int x = 0; x < 3; ++x) {
				temp += *position + " ";
				++position;
			}
			temp += *position + " ";
			for (int x = 0; x < 3; ++x) {
				++position;
				temp += *position + " ";
			}
			for (int x = 0; x < 3; ++x) {
				--position;
			}
			iter = data.find(*position);
			if (iter != data.end()) {
				iter->second += "\n          " + temp;
			}
			else {
				data.insert(pair<string, string>(*position, temp));
			}
		}
```

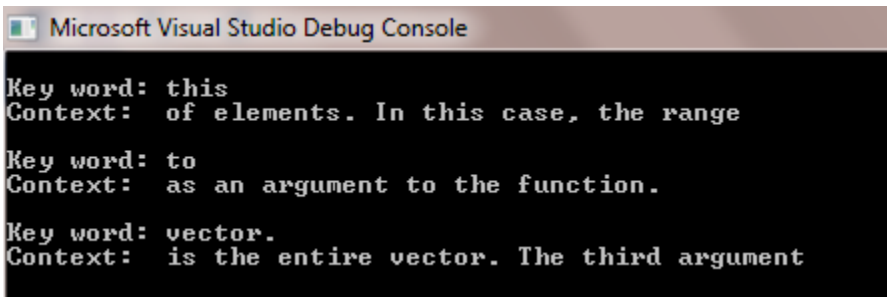
```
                q.pop();
                q.push(*position);
        }

        for (pos = data.begin(); pos != data.end(); ++pos) {
                cout << "Key word: " << pos->first;
                cout << "\nContext: " << pos->second << "\n\n";
        }
}
```

This is the paragraph in the input file, and it is from the textbook:

The first and second arguments specify a range of elements. In this case, the range is the entire vector. The third argument is the name of a function. The for_each algorithm calls the function once for each element in the range, passing the element as an argument to the function.


Microsoft Visual Studio Debug Console
Enter a file name: KeyWords.txt
Key word: In
Context: range of elements. In this case, the
Key word: The
Context: The first and second
the entire vector. The third argument is
of a function. The for_each algorithm calls
Key word: a
Context: second arguments specify a range of elements.
the name of a function. The for_each
Key word: algorithm
Context: function. The for_each algorithm calls the function
Key word: an
Context: the element as an argument to the
Key word: and
Context: The first and second arguments specify
Key word: argument
Context: vector. The third argument is the name
element as an argument to the function.
Key word: arguments
Context: first and second arguments specify a range
Key word: as
Context: passing the element as an argument to
Key word: calls
Context: The for_each algorithm calls the function once
Key word: case,
Context: elements. In this case, the range is
Key word: each
Context: function once for each element in the
Key word: element
Context: once for each element in the range,
range, passing the element as an argument
Key word: elements.
Context: a range of elements. In this case,
Key word: entire
Context: range is the entire vector. The third
Key word: first
Context: The first and second arguments
Key word: for
Context: the function once for each element in


Microsoft Visual Studio Debug Console
Key word: for_each
Context: a function. The for_each algorithm calls the
Key word: function
Context: algorithm calls the function once for each
Key word: function.
Context: name of a function. The for_each algorithm
argument to the function.
Key word: in
Context: for each element in the range, passing
Key word: is
Context: case, the range is the entire vector.
The third argument is the name of
Key word: name
Context: argument is the name of a function.
Key word: of
Context: specify a range of elements. In this
is the name of a function. The
Key word: once
Context: calls the function once for each element
Key word: passing
Context: in the range, passing the element as
Key word: range
Context: arguments specify a range of elements. In
this case, the range is the entire
Key word: range,
Context: element in the range, passing the element
Key word: second
Context: The first and second arguments specify a
Key word: specify
Context: and second arguments specify a range of
Key word: the
Context: In this case, the range is the
the range is the entire vector. The
third argument is the name of a
for_each algorithm calls the function once for
each element in the range, passing the
the range, passing the element as an
an argument to the function.
Key word: third
Context: entire vector. The third argument is the


Microsoft Visual Studio Debug Console
Key word: this
Context: of elements. In this case, the range
Key word: to
Context: as an argument to the function.
Key word: vector.
Context: is the entire vector. The third argument