CS 121 Homework 3

John Ingram

9/25/19

1. **Given the text file write a program that performs the following tasks:**
   1. **Find and count all words that are from 1 to 10 letters long. Print the results to the screen as a table.**

**Code:**

//

// Program1a

//

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

int main(void)

{

ifstream infile;

infile.open("C:/Users/j123j/Documents/GitHub/Sem1FreshmanYear/CS121/Homework3/Program1a/UAH\_sample.txt");

if (!infile)

{

cout << "Error. Not able to open file.";

system("pause");

return(3);

}

cout << setw(15) << "Word Length" << setw(15) << "Count" << endl;

for (int i = 1; i <= 10; i++)

{

int count = 0;

string word;

while (!infile.eof())

{

infile >> word;

int wordLength = word.length();

string digits = "1234567890";

if (word.find(',', 0) < string::npos) wordLength--;

if (word.find('.', 0) < string::npos) wordLength--;

for (int k = 0; k < word.length(); k++)

{

if (digits.find(word[k], 0) < string::npos)

{

count--;

break;

}

}

if (digits.find(word,0))

if (word.length() == i) count++;

}

cout << setw(15) << i << setw(15) << count << endl;

infile.clear();

infile.seekg(0, ios::beg);

}

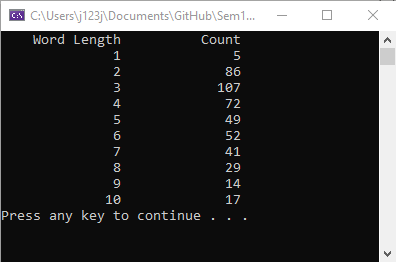
infile.close();

system("pause");

return 0;

}

**Output:**

****

* 1. **Count the number of times each letter (A\_Z, a-z) appears in the file. Print (to the screen) a table of each letter and the number of times it appears. You do not need separate counts for upper and lowercase. Ignore all characters that are not letters.**

**Code:**

//

// Program1b

//

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

int main(void)

{

ifstream infile;

int letterCounts[26];

char currentChar;

for (int i = 0; i < 26; i++) letterCounts[i] = 0;

infile.open("C:/Users/j123j/Documents/GitHub/Sem1FreshmanYear/CS121/Homework3/Program1b/UAH\_sample.txt");

if (!infile)

{

cout << "Error. Not able to open file.";

system("pause");

return(3);

}

while (!infile.eof())

{

infile >> currentChar;

if (isalpha(currentChar)) letterCounts[(int)tolower(currentChar) - 97]++; //add one to the index corresponding to the letter

}

for (int i = 0; i < 26; i++)

{

cout << (char)(i + 97) << ": " << letterCounts[i] << endl;

}

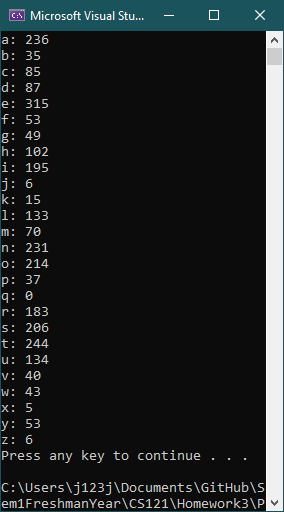
infile.close();

system("pause");

return 0;

}

**Output:**

****

* 1. **Given an input letter find and print to the screen all words that begin with that letter.**

**Code:**

//

// Program1c

//

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

int main(void)

{

ifstream infile;

string word;

char findChar = ' ';

infile.open("C:/Users/j123j/Documents/GitHub/Sem1FreshmanYear/CS121/Homework3/Program1c/UAH\_sample.txt");

if (!infile)

{

cout << "Error. Not able to open file.";

system("pause");

return(3);

}

cout << "Enter the letter you would like to search for: ";

cin >> findChar;

findChar = tolower(findChar);

while (!infile.eof())

{

infile >> word;

if ((isalpha(word[0])) && (tolower(word[0]) == findChar)) cout << word << endl;

}

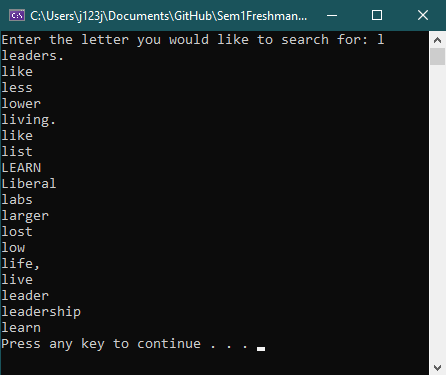
infile.close();

system("pause");

return 0;

}

**Output:**

****

* 1. **Using the original UAH\_sample.txt file replace each occurrence of UAHuntsville with UAH. Create a new file with the updated text.**

**Code:**

//

// Program1d

//

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

int main(void)

{

ifstream infile;

ofstream outfile;

string removed = "UAHuntsville", line;

infile.open("C:/Users/j123j/Documents/GitHub/Sem1FreshmanYear/CS121/Homework3/Program1a/UAH\_sample.txt");

outfile.open("C:/Users/j123j/Documents/GitHub/Sem1FreshmanYear/CS121/Homework3/Program1d/UAH\_updated.txt");

if (!infile)

{

cout << "Error. Not able to open input file.";

system("pause");

return(3);

}

if (!outfile)

{

cout << "Error. Not able to open output file.";

system("pause");

return(4);

}

while (!infile.eof())

{

getline(infile, line);

int i = 0;

do

{

i = line.find(removed, i);

if (i < string::npos) line.erase(i+3, removed.length()-3);

} while (i < string::npos);

outfile << line << endl;

}

infile.close();

outfile.close();

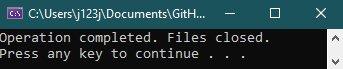
cout << "Operation completed. Files closed." << endl;

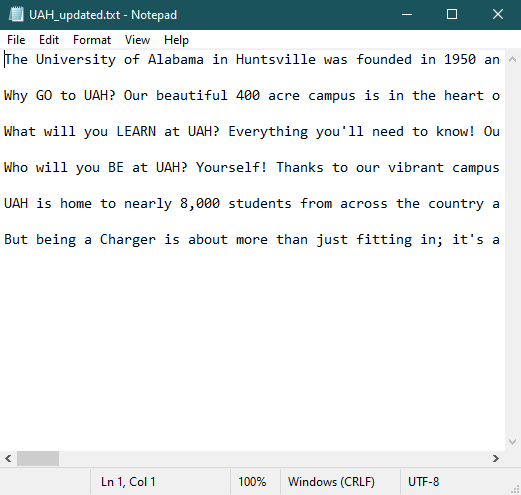
system("pause");

return 0;

}

**Output:**

****

** or see file attached**

1. **Design and write a program that will accept a sentence as input, reverse the words and print the sentence out backwards. The individual words should not be reversed as we did in class. The words in the sentence should be in reverse order.**

**Code:**

//

// Program 2

//

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

int main(void)

{

string input, output = "";

cout << "Enter the text you'd like to flip: ";

getline(cin, input);

for (int i = input.length(); true ; i--)

{

if (input.find(' ', i) != string::npos)

{

output.append(input.substr(input.find(' ', i), input.length()));

input.erase(input.find(' ', i), string::npos);

}

else if (input.find(' ', 0) == string::npos)

{

output.append(" ");

output.append(input);

break;

}

}

output.erase(0, 1);

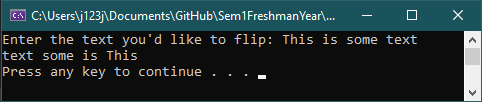
cout << output << endl;

system("pause");

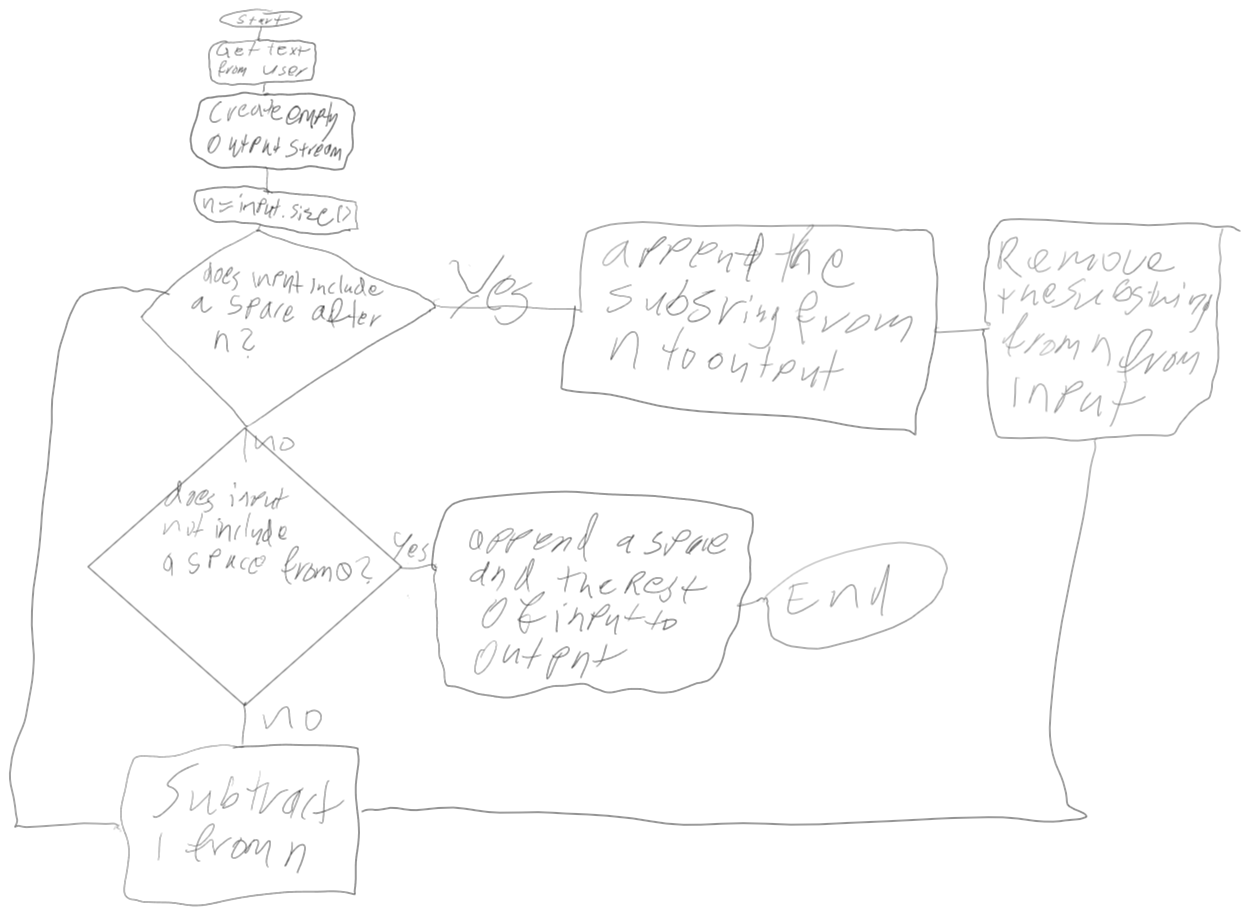
return 0;

}

**Output:**

****

**Diagram:**

****

1. **Write a program that will convert a series of binary digits to their decimal equivalent. For convenience limit the binary number to 16 bits. Write the decimal equivalent to the screen.**

**Code:**

//

// Program 3

//

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

int main(void)

{

string binary;

long long unsigned int result = 0;

cout << "Enter a number in binary to convert it to decimal.\n"

<< "Make sure the result will fit into an unsigned long long int: ";

cin >> binary;

for (int i = 0; i <= binary.length(); i++)

{

if (binary[i] == '1') result += pow(2, binary.length() - (i+1));

}

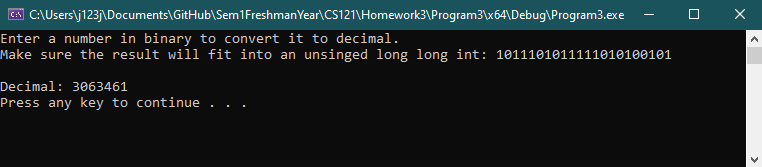
cout << endl << "Decimal: " << result << endl;

system("pause");

return 0;

}

**Output:**

****

1. **Given the word find puzzle below locate and print out all four letter words. The words can appear in their normal form or reversed. There are no vertical or diagonal words – yet. Process one line at a time.**

**Code:**

//

// Program 4

//

#include <iostream>

#include <fstream>

#include <iomanip>

#include <string>

using namespace std;

string reverse(string word);

int main(void)

{

ifstream dictionary, scramble;

string word, line;

char findChar = ' ';

dictionary.open("C:/Users/j123j/Documents/GitHub/Sem1FreshmanYear/CS121/Homework3/Program4/dictionary\_four\_letter\_words.txt");

if (!dictionary)

{

cout << "Error. Not able to open file.";

system("pause");

return(3);

}

scramble.open("C:/Users/j123j/Documents/GitHub/Sem1FreshmanYear/CS121/Homework3/Program4/word\_puzzle.txt");

if (!scramble)

{

cout << "Error. Not able to open file.";

system("pause");

return(3);

}

cout << "Starting..." << endl;

while (!scramble.eof())

{

getline(scramble, line);

while (!dictionary.eof())

{

dictionary >> word;

if (line.find(word) != string::npos || line.find(reverse(word)) != string::npos)

{

cout << word << endl;

}

}

dictionary.clear();

dictionary.seekg(0, ios::beg);

}

dictionary.close();

scramble.close();

cout << "Process complete. Files closed." << endl;

system("pause");

return 0;

}

string reverse(string word)

{

string reversed = "";

for (int i = 0; i < word.length(); i++)

{

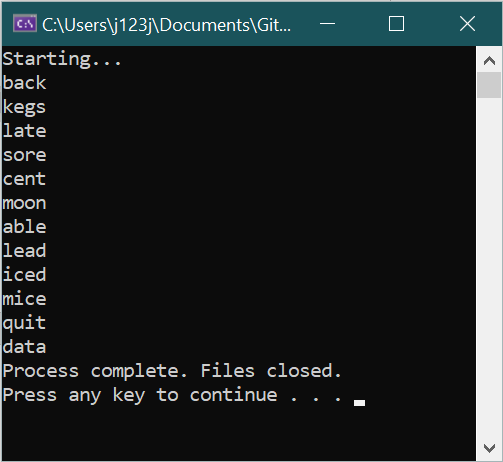
reversed.push\_back(word[i]);

}

return reversed;

}

**Output:**

****