CS32Project3 Report

1. The hardest thing I encountered was thinking about those helper functions for Jumbler function. Like how to print out all the combinations of a word using loop and recursion and how to check if it’s in the dictionary.
2. The ifstream dict file is the one provided by the professor.

I created a smaller dictionary called test.txt in my CS32 project3 folder with elements: ()

A

B

C

D

E

F

U

N

UR

M

A

Ur

dog

dgo

ogd

look

kool

Testing in main: string results[MAXRESULTS];

string dict[MAXDICTWORDS];

ifstream dictfile; // file containing the list of words

**int** nwords; // number of words read from dictionary

string word;

dictfile.open("/Users/haoyuluo/Desktop/CS32/CS32\_Project3/test.txt");

**if** (!dictfile) {

cout << "File not found!" << endl;

**return** (1);

}

nwords = lexiconBuilder(dictfile, dict);

cout << nwords << endl;

cout << "Please enter a string for an anagram: ";

cin >> word;

**int** numMatches = theJumbler(word, dict, nwords, results);

**if** (!numMatches) {

cout << "No matches found" << endl;

}

**else** {

divulgeSolutions(results, numMatches);

}

**return** 0;

test case(string entered for anagram): X // X is not on the list, and test if numMatches is 0 by seeing whether “No matches found is printed out”

test case: dog //A simple test case to see if the program can compile, lexiconbuilder correctly print out the total number of test.txt and the output matching words on the dict correctly.

Test case: look // a test case that test if theJumbler remove duplicated words from the result list.

Test case: change MAXRESULT to 2: input dog again. // Check whether the Jumbler function only put 2 elements into result array and whether divulgeSolutions prints out 2 words.

Test case: change MAXDICTWORDS to 10: input dog again. // Check if lexiconbuilder prints out the correct number which is 10 in this case, and see if the Jumbler function put nothing In the result array and divulgeSolution prints out “no mathc found since noting is in the result.”

Now change dict file to the one provided by professor: “word.txt”

Test case: abuse // A slightly longer word to test if printPermutation print out all the combinations and if the jumbler put the word on dict into result.

Test case: anagram A longer word to test if printPermutation print out all the combinations and if the jumbler put the word on dict into result.