**Name: Oguguom Lawrence C Matric: 230908**

**Department: Computer Science Level: 200L**

**Course: CSC235**

**SOLUTION TO MY ASSIGNMENT**

**WRITING THE PEUSODOCODE ALGORITHM FOR MY CODE ASSIGNMENT**

**INCLUDE <STDIO.H>**

To declare the values of inputs and output

**# INCLUDE <STDLIB.H>**

To include and accept library functions

**INT MAIN()**

Where the whole functions and syntaxes are being written and executed

{

**INT TESTINPUT;**

This is a variable we declared to accept the number of test cases

**CHAR CHECKSTRING[100];**

This is a character variable we declared to accept the length of one hundred characters

**FILE \*IN;**

To open a file from the directory

**PRINTF("PLEASE ENTER TOTAL NUMBER OF INPUT TESTS: ");**

This function is to enable users enter the number of tests that was undertaken

**SCANF("%D", &TESTINPUT);**

And this is to accept whatever test the user enters

if((in = fopen("/home/lawrence/desktop/csc235 assignment2/adam.txt","r")) == NULL){

To bring in the file from the directory so the program can execute what it’s designed for

printf("You Encountered An Error Opening This File");

This pops up only when an error is encountered

//exit

**exit(1);**

Literally it means to terminate and move to the next line of execution

}

**FOR(INT I=1; I <= TESTINPUT;){**

This is the loop statement I used to check my test input

**FSCANF(IN, "%S", TESTINPUT);**

To accept the inputs from the textinput the user enters

**INT READSTEP = 0;**

Creating a variable readstep and assigning zero to it

**FOR(INT J=0; J<100;){**

Creating a loop statement for the string to run from zero to less than a hundred

switch(checkstring[j]){

**CASE 'U':**

readstep += 1;

**j++;**

This would add to the value of J by incrementing it with plus one

**BREAK;**

To terminate it and move to the next line

**CASE 'D':**

**PRINTF("THE TOTAL AMOUNT OF STEPS IN TEST %D IS %D\N", 1, READSTEP );**

To print out the total number of steps and will be in integer because of the **“%d”** syntax

j=100; //i have to terminate the local loop

i++;

break;

default:

**PRINTF("THE STRING CHARACTER IS INVALID");**

This would only work if the character entered is invalid or incorrect

**BREAK;**

To terminate the program after it must have executed our desired instructions

}

}

}

**RETURN 0;**

This is the return statement end function execution and I have it to return the value to the **main** caller

}