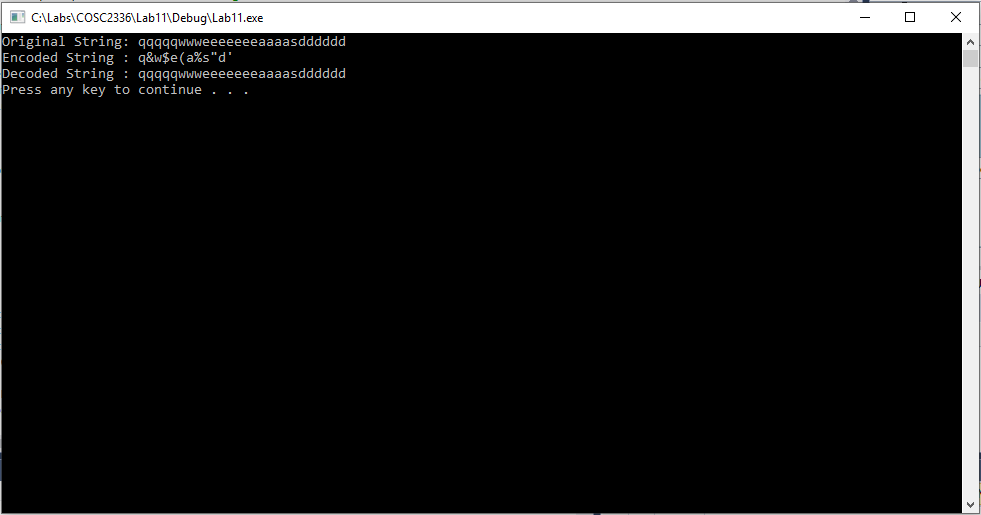
Name: William A. Brannon

Assignment: Lab Assignment Report #11

Date Due: Monday, April, 15, 2019

Class: Fundamentals of Programming III Section #1

**Program Output**



**Lab11.cpp**

// Lab 11: Data Compression

// By: William Brannon on 04/15/2019

#include "stdafx.h"

#include <iostream>

#include <string>

using namespace std;

string simpleRLEencode(string input) {

//When a dataset has many runs and few single characters, it can be more efficient to simply encode everything as though it were a run

//This is the usecase for when the algorithm is optimally efficient.

//It is unwise to use this encoding on consecutive single characters

string output = "";

//output is in the form of character then count

int n = input.length();

for (int i = 0; i < n; i++) {

// Count occurrences of current character

int count = 1;

while (i < n - 1 && input[i] == input[i + 1]) {

count++;

i++;

}

// add character and its count to output

// also convert count to an ascii character so it can encode longer runs as suggested in the textbook

output += input[i];

output += (char)count + 33; //offseting the value because ascii's character set also includes non-visible characters

}

return output;

}

string simpleRLEdecode(string input) {

string output = "";

int n = input.length();

for (int i = 0; i < n / 2; i++) {

int j = input.at(i \* 2 + 1) - 33;

for (int k = 0; k < j; k++) {

output += (input.at(i \* 2));

}

}

return output;

}

int main()

{

string str = "qqqqqwwweeeeeeeaaaasdddddd";

string str2 = simpleRLEencode(str);

string str3 = simpleRLEdecode(str2);

cout << "Original String: " << str << endl;

cout << "Encoded String : " << str2 << endl;

cout << "Decoded String : " << str3 << endl;

system("pause");

return 0;

}