#include <fstream>

#include <iostream>

#include <vector>

using namespace std;

vector<char> Compress(const vector<char>& data) {

vector<char> Compressed;

if (data.empty()) {

return Compressed;

}

char Prev = data[0];

int Count = 1;

for (size\_t i = 1; i < data.size(); i++) {

if (data[i] == Prev) {

Count++;

}

else {

Compressed.push\_back(static\_cast<char>(Count));

Compressed.push\_back(Prev);

Prev = data[i];

Count = 1;

}

}

Compressed.push\_back(static\_cast<char>(Count));

Compressed.push\_back(Prev);

return Compressed;

}

vector<char> DeCompress(const vector<char>& Compressed) {

vector<char> DeCompressed;

for (size\_t i = 0; i < Compressed.size(); i += 2) {

if (i + 1 < Compressed.size()) {

int Count = static\_cast<unsigned char>(Compressed[i]);

char Symbol = Compressed[i + 1];

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

DeCompressed.push\_back(Symbol);

}

}

else {

break;

}

}

return DeCompressed;

}

int main() {

int choice;

do {

cout << "1: Compress File" << endl;

cout << "2: DeCompress File" << endl;

cout << "3: Exit" << endl;

cin >> choice;

if (choice == 1) {

ifstream file("compression.docx", ios::binary);

if (!file) {

cerr << "Error opening input file." << endl;

continue;

}

vector<char> data((istreambuf\_iterator<char>(file)), istreambuf\_iterator<char>());

file.close();

vector<char> Compressed = Compress(data);

ofstream Compressed\_file("Compressed.txt", ios::binary);

if (!Compressed\_file) {

cerr << "Error opening output file." << endl;

continue;

}

Compressed\_file.write(Compressed.data(), Compressed.size());

Compressed\_file.close();

}

else if (choice == 2) {

ifstream Compressed\_file("Compressed.txt", ios::binary);

if (!Compressed\_file) {

cerr << "Error opening compressed file." << endl;

continue;

}

vector<char> Compressed((istreambuf\_iterator<char>(Compressed\_file)), istreambuf\_iterator<char>());

Compressed\_file.close();

vector<char> DeCompressed = DeCompress(Compressed);

ofstream DeCompressed\_file("DeCompressed.docx", ios::binary);

if (!DeCompressed\_file) {

cerr << "Error opening decompressed file." << endl;

continue;

}

DeCompressed\_file.write(DeCompressed.data(), DeCompressed.size());

DeCompressed\_file.close();

}

else if (choice != 3) {

cout << "Invalid Choice" << endl;

}

} while (choice != 3 && !cin.fail());

return 0;

}