AIMEN NADEEM

TASK 3

PROGRAM:

#include<iostream>

#include<string>

#include<fstream>

#include<stdexcept>

using namespace std;

string readfile(string& hello)

{

ifstream file(hello, ios::in | ios::binary);

if (!file) {

throw runtime\_error("File cannot be opened or read");

}

string content((istreambuf\_iterator<char>(file)), istreambuf\_iterator<char>());

return content;

}

void writefile(string& hello, string& info)

{

ofstream file(hello, ios::out | ios::binary);

if (!file)

{

throw runtime\_error("File cannot be opened or read");

}

file.write(info.c\_str(), info.size());

}

string compressRLE(const string& info)

{

string compressd;

int a = info.size();

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

{

int count = 1;

while (i+1<a && info[i] == info[i+1])

{

++i;

++count;

}

compressd += info[i];

compressd += to\_string(count);

}

return compressd;

}

string decompressRLE(const string& info)

{

string decompressd;

int a = info.size();

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

{

char ch = info[i];

string countSTR;

while (i+1 < a && isdigit(info[i+1]))

{

countSTR += info[++i];

}

int count= stoi(countSTR);

decompressd.append(count, ch);

}

return decompressd;

}

int main()

{

string inputhello, outputhello;

char choice;

cout << "enter input filename" << endl;

cin >> inputhello;

cout << "choose option" << endl;

cout << "(c,C) compress file or (d,D) decpmpress file" << endl;

cin >> choice;

if (choice == 'c' || choice == 'C') {

try {

string info = readfile(inputhello);

string compressddata = compressRLE(info);

outputhello = inputhello + ".rle";

writefile(outputhello, compressddata);

cout << "file compressd successfully:" << outputhello << endl;

}

catch (const exception& e) {

cerr << "Error: " << e.what() << endl;

}

}

else if (choice == 'd' || choice == 'D') {

try {

string info = readfile(inputhello);

string decompressddata = decompressRLE(info);

outputhello = inputhello + ".decompressd";

writefile(outputhello, decompressddata);

cout << "file decompressd successfully:" << outputhello << endl;

}

catch (const exception& e) {

cerr << "Error: " << e.what() << endl;

}

}

else {

cout << "invalid choice" << endl;

}

return 0;

}

OUTPUT:

