TASK:03

TITLE: COMPRESSED AND DECOMPLRESSED

FILE

AUTHOR: SIDRA YOUNAS

DATE:23-08-2024

COMPANY: DIGITALEMPOWERMENTNETWORK

Implementing a Simple File Compression Algorithm:

- **Objective:** Develop a basic file compression and decompression tool.
- **Description:** Create a C++ program that reads a file, compresses its content using a simple algorithm (e.g., Run-Length Encoding), and writes the compressed data to a new file. Also, implement decompression.
  Key Steps:
- Reading and writing files
- Implementing the Run-Length Encoding algorithm
- Handling edge cases (e.g., different file types, empty files)
- Creating functions for both compression and decompression

Code Explanation

1. **readFile Function**:
   - Reads the content of a file specified by the `filename`.
   - Opens the file in binary mode and reads it into a string.
2. **writeFile Function**:
   - Writes data to a file specified by `filename`.
   - Opens the file in binary mode and writes the string data.
3. **compressRLE Function**:
   - Compresses a string using RLE by counting consecutive repeating characters.
4. **decompressRLE Function**:
   - Decompresses an RLE-compressed string by expanding each character based on its associated count.
5. **main Function**:
   - Asks the user to choose between compressing or decompressing.
   - Prompts for the input and output file names.
   - Reads the input file, processes the content based on the user's choice, and writes the result to the output file.

Code:

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

// Function to read the content of a file
string readFile(const string& filename) {
    ifstream file(filename, ios::binary);
    if (!file.is_open()) {
        cerr << "Error opening file: " << filename << endl;
        return "";
    }
```

```
    string content((istreambuf_iterator<char>(file)), istreambuf_iterator<char>());
    return content;
}

// Function to write data to a file
void writeFile(const string& filename, const string& data) {
    ofstream file(filename, ios::binary);
    if (!file.is_open()) {
        cerr << "Error opening file for writing: " << filename << endl;
        return;
    }
    file.write(data.c_str(), data.size());
}

// Function to compress the input using Run-Length Encoding
string compressRLE(const string& input) {
    string compressed;
    int length = input.length();

    for (int i = 0; i < length; i++) {
        int count = 1;
        while (i < length - 1 && input[i] == input[i + 1]) {
            count++;
            i++;
        }
        compressed += input[i];
        compressed += to_string(count);
    }
    return compressed;
}

// Function to decompress the input using Run-Length Encoding
string decompressRLE(const string& input) {
    string decompressed;
    int length = input.length();

    for (int i = 0; i < length; i++) {
        char currentChar = input[i];
        string countStr;
        while (isdigit(input[++i])) {
            countStr += input[i];
        }
        i--;
        int count = stoi(countStr);
        decompressed.append(count, currentChar);
    }
    return decompressed;
}

int main() {
    string inputFile, outputFile;
    int choice;

    cout << "Enter 1 to compress or 2 to decompress: ";
    cin >> choice;
    cout << "Enter the input file name: ";
    cin >> inputFile;
    cout << "Enter the output file name: ";
```

```
    cin >> outputFile;

    string content = readFile(inputFile);

    if (content.empty()) {
        cerr << "Failed to read the file. Exiting program.\n";
        return 1; // Non-zero exit code to indicate failure
    }

    if (choice == 1) {
        string compressedContent = compressRLE(content);
        writeFile(outputFile, compressedContent);
        cout << "File compressed successfully.\n";
    }
    else if (choice == 2) {
        string decompressedContent = decompressRLE(content);
        writeFile(outputFile, decompressedContent);
        cout << "File decompressed successfully.\n";
    }
    else {
        cerr << "Invalid choice!\n";
    }

    return 0;
}
```

Output:

```
Enter 1 to compress or 2 to decompress: 1
Enter the input file name: input.txt
Enter the output file name: compressed.txt
```

```
File compressed successfully.
```

```
Enter 1 to compress or 2 to decompress: 2
Enter the input file name: compressed.txt
Enter the output file name: decompressed.txt
```

```
File decompressed successfully.
```