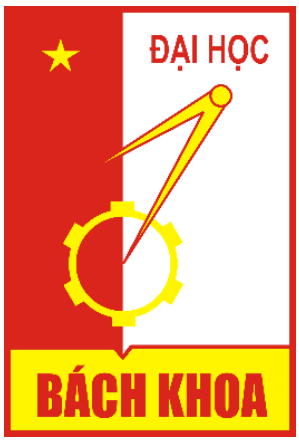
ĐẠI HỌC BÁCH KHOA HÀ NỘI

Trường Công nghệ Thông tin và Truyền thông

----- ☙ 🕮 ❧ -----



Báo cáo Project 1

Đề tài: Thuật toán mã hóa và giải mã RLC

Giảng viên: **PGS.TS. Đặng Văn Chuyết**

Nhóm sinh viên thực hiện:

STT	Họ và tên	MSSV
1	Nguyễn Tiến Thành	20215640
2	Hà Văn Tăng	20215638

Hà Nội, năm 2023

MỤC LỤC

MỞ ĐẦU

Ngày nay, trong bối cảnh phát triển vượt bậc của công nghệ thông tin, vấn đề về tối ưu hóa không gian lưu trữ và hiệu suất truyền là rất quan trọng. Mỗi ngày, chúng ta gặp phải lượng lớn dữ liệu cần phải được truyền tải và lưu trữ một cách hiệu quả. Để đối mặt với thách thức này, các phương pháp mã hóa dữ liệu trở nên ngày càng quan trọng. Một trong những phương pháp nén dữ liệu đơn giản nhưng hiệu quả là thuật toán mã hóa và giải mã Run-Length Encoding (RLC).

Chúng em đã phát triển một chương trình thực hiện mã hóa văn bản đầu vào thành dạng nén và sau đó giải mã nó để khôi phục lại văn bản gốc. Chương trình sẽ bắt đầu với việc mã hóa và giải mã RLC cho các bản tin chứa văn bản không có chứa số. Sau đó, chúng ta sẽ mở rộng chương trình để xử lý trường hợp bản tin có chứa số. Mục tiêu chính của dự án này là để tìm hiểu rõ về thuật toán RLC cơ bản, triển khai nó bằng ngôn ngữ lập trình C++, đánh giá hiệu suất của chương trình mã hóa và giải mã trong các tình huống khác nhau.

Chúng em hi vọng rằng báo cáo này sẽ đưa ra cái nhìn toàn diện về quá trình mã hóa và giải mã RLC sử dụng ngôn ngữ lập trình C++ và có thể hữu ích cho những người quan tâm đến lĩnh vực nén dữ liệu.

Trân trọng!

CHƯƠNG I. Giới thiệu bài toán

I.1. Thuật toán RLC – Run Length Coding

Thuật toán RLC là một phương pháp nén dữ liệu đơn giản bằng cách thay thế loạt các ký tự lặp lại bằng một ký tự và số lần lặp lại của nó. Thuật toán này giúp giảm kích thước của dữ liệu mà vẫn giữ được thông tin quan trọng.

I.2. Bài toán cụ thể

Thuật toán mã hóa và giải mã RLC cho một bản tin như sau:

Thuật toán mã hóa: Tìm trong bản tin những loạt tin giống hệt nhau và thay nó bằng 1 tin trong loạt và độ dài của loạt để tạo ra bản tin bị nén

Thuật toán giải mã: Tìm vị trí là độ dài của loạt và phục hồi lại loạt tin giống nhau có độ dài tìm được

Nếu trong bản tin có chứa các số thì trước vị trí của độ dài loạt của bản tin bị nén cần chèn 1 ký tự đặc biêt là ký tự DLE trong bảng mã ASCII

a. Hãy viết chương trình cho phép nhập một bản tin là đoạn văn bản không chứa các số và sử dụng thuật toán mã hóa RLC để mã hóa nó
b. Sử dụng thuật toán giải mã để phục hồi lại bản tin ban đầu từ bản tin đã bị nén bởi câu a.
c. Thực hiện lại câu a và b với đoạn văn bản có chứa số

CHƯƠNG II. Phân tích bài toán

II.1. Thuật toán mã hóa RLC với file text không chứa chữ số

Đầu vào : File text chứa văn bản không có chữ số

Đầu ra : File text chứa văn bản đã được mã hóa.

Thuật toán RLC :

- Duyệt từng dòng trong file text
- Với mỗi dòng, kiểm tra xem có chứa chữ số hay không. Nếu có thì dừng lại và thông báo file có chứa chữa số. Ngược lại thì làm bước tiếp theo
- Duyệt qua từng ký tự của dòng đó để xác định các loạt tin giống nhau, xác định độ dài của loạt tin
- Nếu độ dài loạt tin lớn hơn 2 thì thay thế loạt tin bằng độ dài của loạt tin và ký tự bị lặp lại. Ngược lại thì giữ nguyên.
- Lưu kết quả thu được vào file text đầu ra chính là file mã hóa
- Tiến hành tính toán kích thước file gốc, kích thước file giải mã, tỷ lệ nén file và thời gian thực thi.

II.2. Thuật toán giải mã RLC với file text không chứa chữ số

Đầu vào : File text chứa văn bản đã được nén từ một văn bản không chứa số

Đầu ra : File text chứa văn bản đã được giải mã

Thuật toán RLC :

- Duyệt từng dòng trong file text
- Với mỗi dòng, duyệt qua từng ký tự của dòng đó để xác định được độ dài của loạt tin đã được mã hóa và ký tự được mã hóa. Tiến hành khôi phục lại loạt tin.
- Lưu kết quả thu được và file text đầu ra chính là file giải mã
- Tiến hành tính toán kích thước file gốc, kích thước file giải mã, tỷ lệ giải nén file và thời gian thực thi.

II.3. Thuật toán mã hóa RLC với file text có chứa chữ số

Đầu vào : File text chứa văn bản có chữ số

Đầu ra : File text chứa văn bản đã được mã hóa.

Thuật toán RLC :

- Duyệt từng dòng trong file text

- Duyệt qua từng ký tự của dòng đó để xác định các loạt tin giống nhau, xác định độ dài của loạt tin, với quy ước độ dài loạt tin nhỏ hơn 10. Khi độ dài đạt tới 9 thì cần thực hiện xử lý luôn.
- Nếu độ dài loạt tin lớn hơn 2 thì thay thế loạt tin bằng xâu kết hợp giữa ký tự dle, độ dài của loạt tin và ký tự bị lặp lại, tiếp tục duyệt dòng đó từ vị trí hiện tại. Ngược lại thì giữ nguyên.
- Lưu kết quả thu được vào file text đầu ra chính là file mã hóa
- Tiến hành tính toán kích thước file gốc, kích thước file giải mã, tỷ lệ nén file và thời gian thực thi.

II.4. Thuật toán giải mã RLC với file text có chứa chữ số

Đầu vào : File text chứa văn bản đã được nén từ một văn bản có chứa số

Đầu ra : File text chứa văn bản đã được giải mã

Thuật toán RLC :

- Duyệt từng dòng trong file text
- Với mỗi dòng, duyệt qua từng ký tự của dòng đó để xác định được độ dài của loạt tin đã được mã hóa và ký tự được mã hóa. Độ dài loạt tin có giá trị từ 3 đến 9, luôn đứng sau ký tự dle và đứng trước ký tự được mã hóa. Tiến hành khôi phục lại loạt tin.
- Lưu kết quả thu được và file text đầu ra chính là file giải mã
- Tiến hành tính toán kích thước file gốc, kích thước file giải mã, tỷ lệ giải nén file và thời gian thực thi.

CHƯƠNG III. Xây dựng chương trình

Chương trình mã hóa và giải mã RLC được xây dựng để thực hiện quá trình nén và giải nén dữ liệu văn bản với bốn chức năng chính :

1. Mã hóa RLC với file text không chứa chữ số
2. Giải mã RLC với file text được mã hóa từ file text không chứa chữ số
3. Mã hóa RLC với file text có chứa chữ số
4. Giải mã RLC với file text được mã hóa từ file text có chứa chữ số

Chương trình cũng cung cấp Menu linh hoạt cho phép người dùng lựa chọn sử dụng các chức năng trên, cũng như các thông tin bên cạnh như kích thước các file, hiệu suất, thời gian thực thi.

III.1. Hàm mã hóa RLC với file text không chứa chữ số

```
23  // Mã hóa RLC với xâu không chứa chữ số
24  string rlc_encoding_without_number(const string& input){
25      string out ="";
26      if(check(input)){
27
28          int i = 0;
29          while(i < input.length()){
30              int count = 1;
31              while(i + 1 < input.length() && input[i] == input[i+1]){
32                  count ++;
33                  i++;
34              }
35              if(count >= 3){
36                  out += to_string(count) + input[i];
37              }else{
38                  for(int q = 0; q < count ; q++){
39                      out += input[i];
40                  }
41
42              }
43              i++;
44          }
45          return out;
46      }
47      return out;
48
49  }
```

Hàm mã hóa RLC với xâu không chứa chữ số

```cpp
// Tiến hành mã hóa với file text không chứa chữ số.
// sử dụng hàm rlc_encoding_without_number trên từng dòng
void rlc_encoding_file_without_number() {
    string inputFile;
    string ouputFile;

    cout <<"Nhập tên file bạn muốn nén: "        ;
    cin >> inputFile;
    cout << "Nhập tên file bạn muốn xuất: "         ;
    cin >> ouputFile;

    ifstream input(inputFile);
    ofstream output(ouputFile);
    if(input.is_open() && output.is_open()){

        // Bắt đầu đo thời gian
        auto start = std::chrono::high_resolution_clock::now();

        string line;
        int k ;
        while(getline(input,line)){
            if(check(line)){
                string outpuLine = rlc_encoding_without_number(line);
                output << outpuLine << endl;
            }else{
                cout<<"\nMã hóa bị dừng do văn bản có chứa chữ số."                << endl;
                system("pause");
                return;
            }


        }
        // Kết thúc đo thời gian
        auto end = std::chrono::high_resolution_clock::now();

        // Tính thời gian đã trôi qua
        std::chrono::duration<double> elapsed = end - start;

        //cout <<"Mã hóa thành công." << endl;
        ifstream fileIn(inputFile,ios::binary);
        fileIn.seekg(0,ios::end);
        streampos fileInSize = fileIn.tellg();

        ifstream fileOu(ouputFile,ios::binary);
        fileOu.seekg(0,ios::end);
        streampos fileOuSize = fileOu.tellg();

        cout <<"\nKích thước file gốc: "      <<formatFileSize(fileInSize) << endl;
        cout <<"Kích thước file nén: "     <<formatFileSize(fileOuSize) << endl;

        double fileSizeIn = static_cast<double>(fileInSize);
        double fileSizeOut = static_cast<double>(fileOuSize);

        double HieuSuat = (fileSizeOut / fileSizeIn) * 100.0;

        cout << "Hệ số nén file: "      << HieuSuat << " %" << endl;
        cout <<"Thời gian thực thi: "     <<elapsed.count() <<" giây.\n"  << endl;

        input.close();
        output.close();


    }else{
        cout <<"\nThất bại do không thể mở file\n"          ;
    }
    system("pause");
}
```

Hàm mã hóa RLC với file text không chứa chữ số

III.2. Hàm giải mã RLC với file text được mã hóa từ file text không chứa chữ số

```
52
53  // Giải mã xâu được mã hóa từ một xâu không chứa chữ số
54  string rlc_decoding_without_number(const string& input){
55      string decoded_data = "";
56      int i = 0;
57      while (i < input.length()) {
58          long count = 0;
59          while (isdigit(input[i])) {
60              count = count * 10 + (input[i] - '0');
61              i++;
62          }
63          if (count > 0) {
64              char current_char = input[i];
65              for (int j = 0; j < count; j++) {
66                  decoded_data += current_char;
67              }
68          } else {
69              decoded_data += input[i];
70          }
71          i++;
72      }
73      return decoded_data;
74  }
```

Hàm giải mã RLC với văn bản được mã hóa từ văn bản không chứa số

```cpp
//Tiến hành giải mã với file text đã được mã hóa từ một file text không chứa chữ số
// sử dụng hàm rlc_decoding_without_number trên từng dòng
void rlc_decoding_file_without_number() {
    string inputFile;
    string ouputFile;

    cout <<"Nhập tên file bạn muốn giải mã: "          ;
    cin >> inputFile;
    cout << "Nhập tên file bạn muốn xuất: "         ;
    cin >> ouputFile;

    ifstream input(inputFile);
    ofstream output(ouputFile);
    if(input.is_open() && output.is_open()){

        // Bắt đầu đo thời gian
        auto start = std::chrono::high_resolution_clock::now();

        string line;
        while(getline(input,line)){

            string outpuLine = rlc_decoding_without_number(line);
            output << outpuLine << endl;

        }
        // Kết thúc đo thời gian
        auto end = std::chrono::high_resolution_clock::now();

        // Tính thời gian đã trôi qua
        std::chrono::duration<double> elapsed = end - start;

        cout <<"\nGiải mã  thành công."      << endl;
        ifstream fileIn(inputFile,ios::binary);
        fileIn.seekg(0,ios::end);
        streampos fileInSize = fileIn.tellg();

        ifstream fileOu(ouputFile,ios::binary);
        fileOu.seekg(0,ios::end);
        streampos fileOuSize = fileOu.tellg();

        cout <<"\nKích thước file gốc: "      <<formatFileSize(fileInSize) << endl;
        cout <<"Kích thước file giải mã: "       <<formatFileSize(fileOuSize) << endl;

        double fileSizeIn = static_cast<double>(fileInSize);
        double fileSizeOut = static_cast<double>(fileOuSize);

        double HieuSuat = (fileSizeOut / fileSizeIn) * 100.0;

        cout << "Tỉ lệ giải nén: "        << HieuSuat << " %" << endl;
        cout <<"Thời gian thực thi: "     <<elapsed.count() <<" giây.\n"  << endl;

        input.close();
        output.close();


    }else{
        cout <<"\nThất bại do không thể mở file\n"         ;
    }
    system("pause");
}
```

Hàm giải mã RLC với file text được mã hóa từ file text không chứa số

III.3. Hàm mã hóa RLC với file text chứa chữ số

```
76  // Mã hóa RLC với xâu có chứa chữ số
77  string rlc_encoding_with_number(const string& input){
78      char dle = 0x10;
79      string out ="";
80      int i = 0;
81      while(i < input.length()){
82              int count = 1;
83              while(i +1< input.length() && input[i] == input[i+1]){
84                  count ++;
85                  i++;
86                  if(count == 9){
87                      out +=  dle +  to_string(count) + input[i];
88                      count = 0;
89                  }
90              }
91              if(count >= 3){
92                  out += dle+ to_string(count) + input[i];
93              }else{
94                  for(int q = 0; q < count ; q++){
95                      out += input[i];
96                  }
97              }
98              i++;
99          }
00      return out;
01  }
```

Hàm mã hóa RLC với văn bản có chứa chữ số

```
269
270 // Tiến hành mã hóa với file text có chứa chữ số
271 // sử dụng hàm rlc_encoding_with_number trên từng dòng
272 void rlc_encoding_file_with_number() {
273     string inputFile;
274     string ouputFile;
275
276     cout <<"Nhập tên file bạn muốn nén: "        ;
277     cin >> inputFile;
278     cout << "Nhập tên file bạn muốn xuất: "         ;
279     cin >> ouputFile;
280
281     ifstream input(inputFile);
282     ofstream output(ouputFile);
283     if(input.is_open() && output.is_open()){
284
285         // Bắt đầu đo thời gian
286         auto start = std::chrono::high_resolution_clock::now();
287
288         string line;
289         while(getline(input,line)){
290
291                 string outpuLine = rlc_encoding_with_number(line);
292                 output << outpuLine << endl;
293
294
295         }
296         // Kết thúc đo thời gian
297         auto end = std::chrono::high_resolution_clock::now();
298
299         // Tính thời gian đã trôi qua
300         std::chrono::duration<double> elapsed = end - start;
301
302         cout <<"Mã hóa thành công."     << endl;
303         ifstream fileIn(inputFile,ios::binary);
304         fileIn.seekg(0,ios::end);
305         streampos fileInSize = fileIn.tellg();
306
307         ifstream fileOu(ouputFile,ios::binary);
308         fileOu.seekg(0,ios::end);
309         streampos fileOuSize = fileOu.tellg();
310
311         cout <<"\nKích thước file gốc: "      <<formatFileSize(fileInSize) << endl;
312         cout <<"Kích thước file nén: "     <<formatFileSize(fileOuSize) << endl;
313
314         double fileSizeIn = static_cast<double>(fileInSize);
315         double fileSizeOut = static_cast<double>(fileOuSize);
316
317         double HieuSuat = (fileSizeOut / fileSizeIn) * 100.0;
318
319         cout << "Hệ số nén file: "      << HieuSuat << " %" << endl;
320         cout <<"Thời gian thực thi: "     <<elapsed.count() <<" giây.\n"  << endl;
321
322
323         input.close();
324         output.close();
325
326     }else{
327         cout <<"\nThất bại do không thể mở file\n"         ;
328     }
329     system("pause");
330 }
```

Hàm mã hóa RLC với file text có chứa chữ số

III.4. Hàm giải mã RLC với file text được mã hóa từ file text có chứa chữ số

```
104  //Giải mã xâu đã mã hóa từ một xâu có chứa chữ số
105  string rlc_decoding_with_number(const string& input){
106      char dle = 0x10;
107      string out ="";
108      int i = 0;
109      while(i < input.length()){
110          if(input[i] == dle){
111              int count = input[i+1] - '0';
112
113              for( int j = 0; j < count-1 ; j++){
114                  out += input[i+2];
115              }
116              i+=2;
117          }
118          out += input[i];
119          i++;
120      }
121      return out;
122  }
123
```

Hàm giải mã RLC với văn bản được mã hóa từ văn bản có chứa chữ số

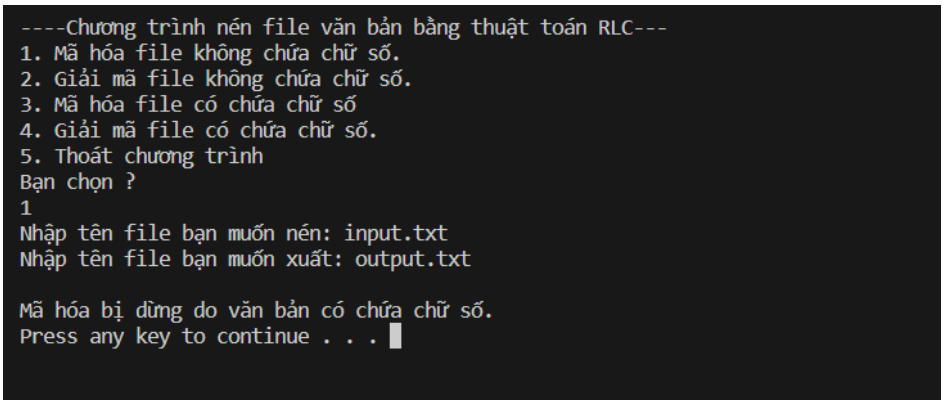
```cpp
// Tiến hành giải mã với file text được mã hóa từ file text có chứa chữ số
// sử dụng hàm rlc_decoding_with_number trên từng dòng
void rlc_decoding_file_with_number() {
    string inputFile;
    string ouputFile;

    cout <<"Nhập tên file bạn muốn giải mã: "          ;
    cin >> inputFile;
    cout << "Nhập tên file bạn muốn xuất: "         ;
    cin >> ouputFile;

    ifstream input(inputFile);
    ofstream output(ouputFile);
    if(input.is_open() && output.is_open()){

        // Bắt đầu đo thời gian
        auto start = std::chrono::high_resolution_clock::now();

        string line;
        while(getline(input,line)){

            string outpuLine = rlc_decoding_with_number(line);
            output << outpuLine << endl;

        }
        // Kết thúc đo thời gian
        auto end = std::chrono::high_resolution_clock::now();

        // Tính thời gian đã trôi qua
        std::chrono::duration<double> elapsed = end - start;

        cout <<"\nGiải mã  thành công."      << endl;
        ifstream fileIn(inputFile,ios::binary);
        fileIn.seekg(0,ios::end);
        streampos fileInSize = fileIn.tellg();

        ifstream fileOu(ouputFile,ios::binary);
        fileOu.seekg(0,ios::end);
        streampos fileOuSize = fileOu.tellg();

        cout <<"\nKích thước file gốc: "      <<formatFileSize(fileInSize) << endl;
        cout <<"Kích thước file giải mã: "       <<formatFileSize(fileOuSize) << endl;

        double fileSizeIn = static_cast<double>(fileInSize);
        double fileSizeOut = static_cast<double>(fileOuSize);

        double HieuSuat = (fileSizeOut / fileSizeIn) * 100.0;

        cout << "Tỉ lệ giải nén: "        << HieuSuat << " %" << endl;
        cout <<"Thời gian thực thi: "     <<elapsed.count() <<" giây.\n"  << endl;

        input.close();
        output.close();


    }else{
        cout <<"\nThất bại do không thể mở file\n"         ;
    }
    system("pause");
}
```

Hàm giải mã RLC với file text được mã hóa từ file có chứa chữ số

CHƯƠNG IV. Kiểm thử chương trình

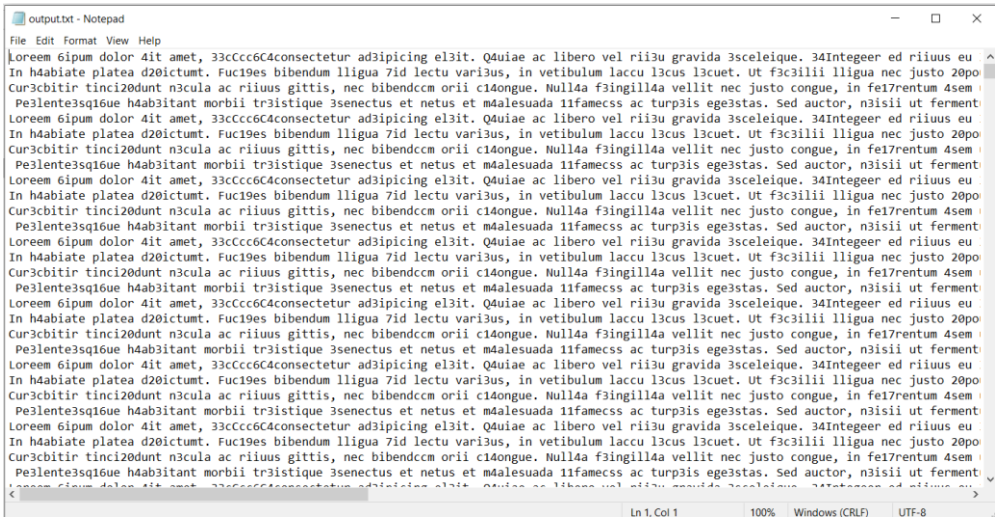
IV.1. Kiểm thử chức năng mã hóa file text không chứa chữ số

-Trường hợp 1: file input.txt có chứa chữ số

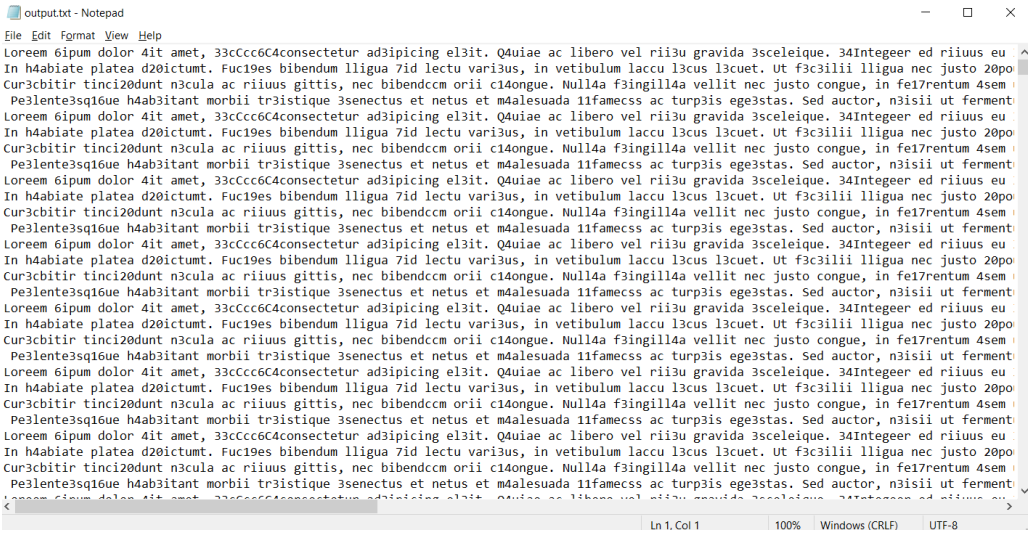


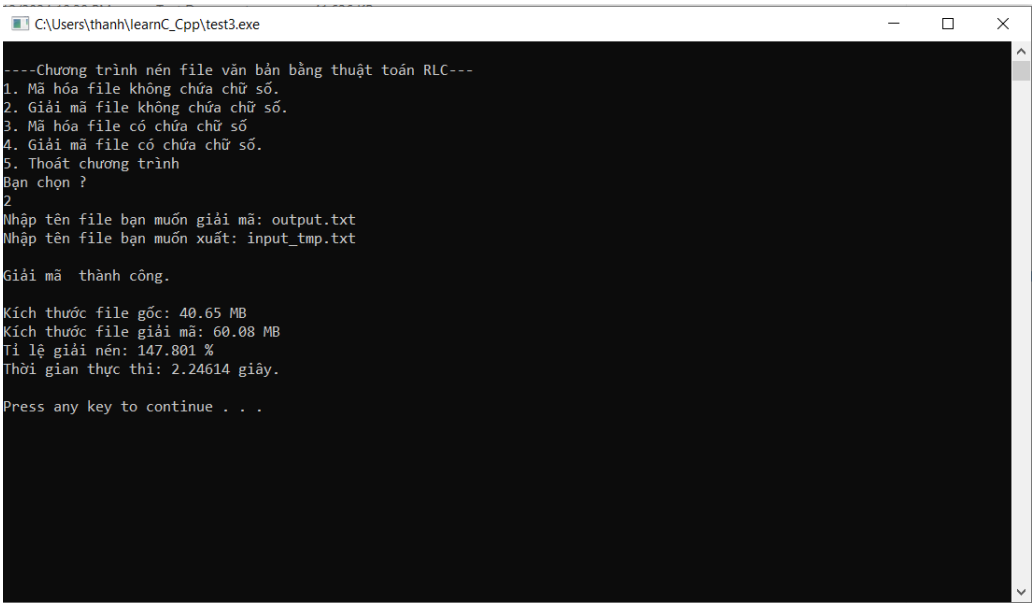
-Trường hợp 2: file input.txt không chứa chữ số

```
----Chương trình nén file văn bản bằng thuật toán RLC---
1. Mã hóa file không chứa chữ số.
2. Giải mã file không chứa chữ số.
3. Mã hóa file có chứa chữ số
4. Giải mã file có chứa chữ số.
5. Thoát chương trình
Bạn chọn ?
1
Nhập tên file bạn muốn nén: input.txt
Nhập tên file bạn muốn xuất: output.txt

Kích thước file gốc: 60.08 MB
Kích thước file nén: 40.65 MB
Hệ số nén file: 67.6586 %
Thời gian thực thi: 2.67516 giây.

Press any key to continue . . .
```

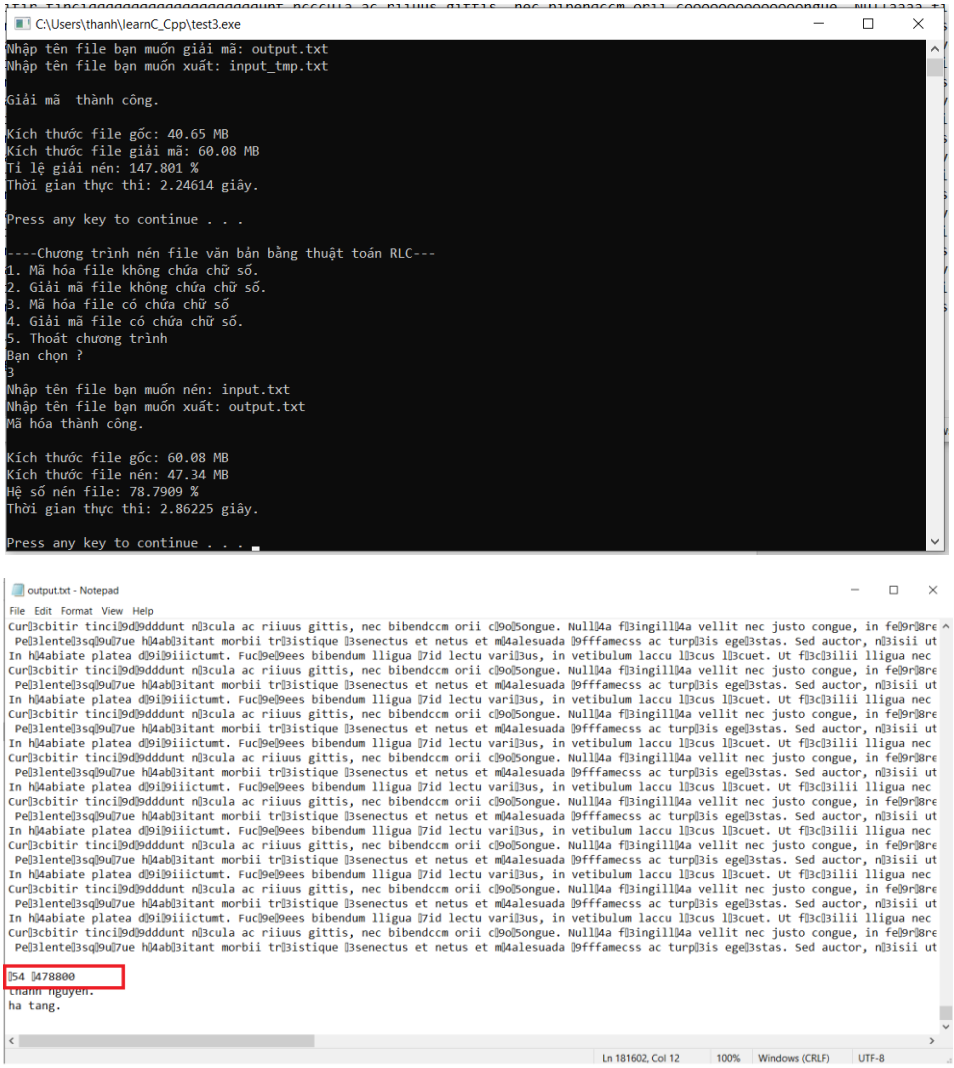


IV.2. Kiểm thử chức năng giải mã file text được mã hóa từ file text không chứa chữ số

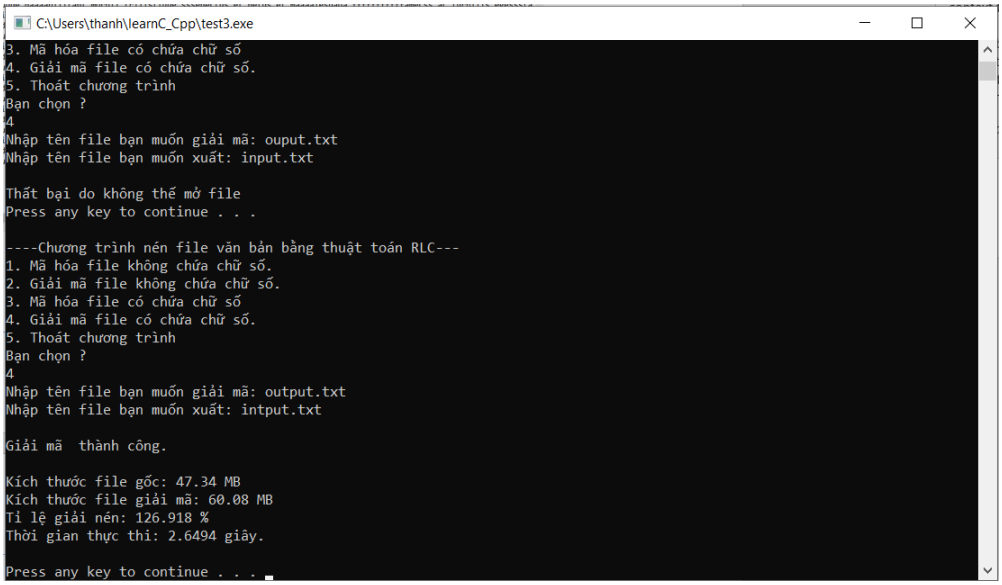


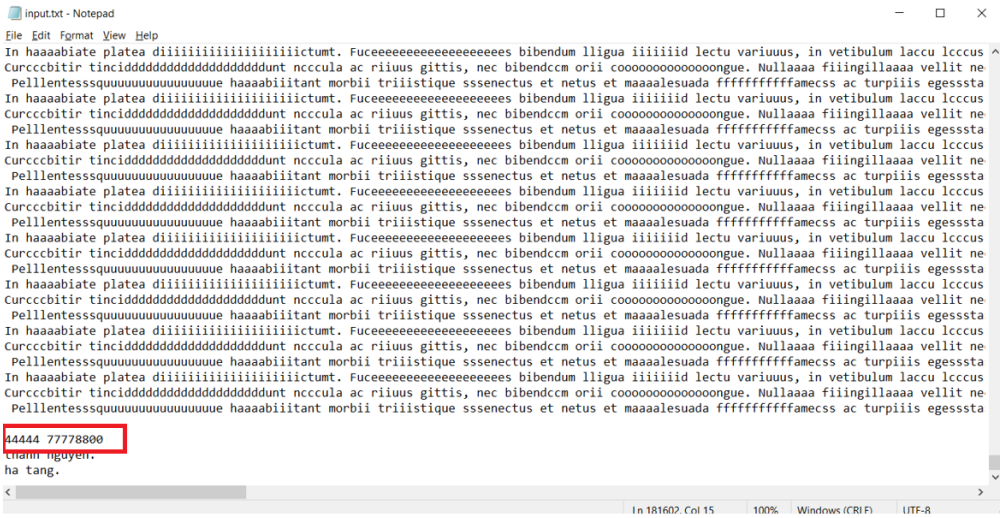


IV.3. Kiểm thử chức năng mã hóa file text có chứa chữ số



IV.4. Kiểm thử chức năng giải mã file text được mã hóa từ file text có chứa chữ số


C:\Users\thanh\learnC_Cpp\test3.exe
3. Mã hóa file có chứa chữ số
4. Giải mã file có chứa chữ số.
5. Thoát chương trình
Bạn chọn ?
4
Nhập tên file bạn muốn giải mã: ouput.txt
Nhập tên file bạn muốn xuất: input.txt
Thất bại do không thể mở file
Press any key to continue . . .
----Chương trình nén file văn bản bằng thuật toán RLC---
1. Mã hóa file không chứa chữ số.
2. Giải mã file không chứa chữ số.
3. Mã hóa file có chứa chữ số
4. Giải mã file có chứa chữ số.
5. Thoát chương trình
Bạn chọn ?
4
Nhập tên file bạn muốn giải mã: output.txt
Nhập tên file bạn muốn xuất: intput.txt
Giải mã thành công.
Kích thước file gốc: 47.34 MB
Kích thước file giải mã: 60.08 MB
Tỉ lệ giải nén: 126.918 %
Thời gian thực thi: 2.6494 giây.
Press any key to continue . . .


input.txt - Notepad
File Edit Format View Help
44444 77778800
thanh nguyen.
ha tang.

KẾT LUẬN

Như vậy, chương trình mã hóa RLC mà chúng em phát triển đã thành công trong việc thực hiện các chức năng, nhiệm vụ mã hóa và giải mã dữ liệu dạng văn bản dựa trên thuật toán RLC – Run-Length Coding. Chương trình có khả năng xử lý và nén giải dữ liệu đáng kể khi có sự lặp lại liên tiếp của các ký tự trong file đầu vào, cũng như cung cấp chức năng giải mã file về dạng ban đầu.

Qua đây, ta hiểu được một cách tổng quan, cơ bản về phương thức hoạt động của thuật toán mã hóa RLC, thấy được sự hiệu quả của thuật trong việc giảm kích thước dữ liệu dạng văn bản cũng như tiềm năng cải tiến trong tương lai.

PHỤ LỤC

Mã nguồn của chương trình : https://github.com/thanhf47/project1