ĐẠI HỌC BÁCH KHOA HÀ NỘI Trường Công nghệ Thông tin và Truyền thông

---- 80 D cs ----



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:

STTHọ và tênMSSV1Nguyễn Tiến Thành202156402Hà Văn Tăng20215638

Hà Nội, năm 2023

MỤC LỤC

MỞ ĐẦU	3
CHƯƠNG I. Giới thiệu bài toán	4
I.1. Thuật toán RLC – Run Length Coding	4
I.2. Bài toán cụ thể	4
CHƯƠNG II. Phân tích bài toán	5
II.1. Thuật toán mã hóa RLC với file text không chứa chữ số	5
II.2. Thuật toán giải mã RLC với file text không chứa chữ số	5
II.3. Thuật toán mã hóa RLC với file text có chứa chữ số	5
II.4. Thuật toán giải mã RLC với file text có chứa chữ số	6
CHƯƠNG III. Xây dựng chương trình	7
III.1. Hàm mã hóa RLC với file text không chứa chữ số	7
III.2. Hàm giải mã RLC với file text được mã hóa từ file text không c số	
III.3. Hàm mã hóa RLC với file text chứa chữ số	11
III.4. Hàm giải mã RLC với file text được mã hóa từ file text có chứa	chữ số . 13
CHƯƠNG IV. Kiểm thử chương trình	15
IV.1. Kiểm thử chức năng mã hóa file text không chứa chữ số	15
IV.2. Kiểm thử chức năng giải mã file text được mã hóa từ file text kl	
IV.3. Kiểm thử chức năng mã hóa file text có chứa chữ số	17
IV.4. Kiểm thử chức năng giải mã file text được mã hóa từ file text có số	
KÉT LUẬN	20
PHU LUC	21

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 trong!

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 biết 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 đô dài của loat 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
26 🖃
          string out ="
          if(check(input)){
              int i = 0;
while(i < input.length()){</pre>
28
29日
30
                  int count = 1;
while(i + 1 < input.length() && input[i] == input[i+1]){</pre>
31 🖨
32
33
                       count ++;
                       i++:
35 🖨
                   if(count >= 3){
                       out += to_string(count) + input[i];
36
37
                   }else{
                       for(int q = 0; q < count ; q++){
39
40
                            out += input[i];
41
42
43
                   i++;
44
45
              return out;
46
47
          return out;
48
49 L }
```

Hàm mã hóa RLC với xâu không chứa chữ số

```
149 // Tiến hành mã hóa với file text không chứa chữ số.
150 // sử dụng hàm rlc_encoding_without_number trên từng
151 void rlc_encoding_file_without_number() {
                                                 ber trên từng dòng
152
          string inputFile;
153
154
          string ouputFile;
          cout <<"Nhập tên file bạn muốn nén: "
cin >> inputFile;
cout << "Nhập tên file bạn muốn xuất: "
cin >> ouputFile;
155
156
157
158
159
          ifstream input(inputFile);
160
161
          ofstream output(ouputFile);
162 🗀
          if(input.is_open() && output.is_open()){
               // Bắt đầu đo thời gian
auto start = std::chrono::high_resolution_clock::now();
164
165
166
167
               string line;
               int k;
while(getline(input,line)){
168
169
170
                   if(check(line)){
                       cneck(line);
string outpuLine = rlc_encoding_without_number(line);
output << outpuLine << endl;</pre>
171
172
173
                   }else{
                       cout<<"\nMã hóa bị dừng do văn bản có chứa chữ số."
system("pause");
174
                                                                                                 << endl:
175
176
177
                        return
178
179
180
181
                 // Kết thúc đo thời gian
182
                 auto end = std::chrono::high_resolution_clock::now();
183
184
185
                  // Tính thời gian đã trôi qua
186
                 std::chrono::duration<double> elapsed = end - start;
187
188
                  //cout <<"Mã hóa thành công." << endl;
                 ifstream fileIn(inputFile,ios::binary);
189
190
                 fileIn.seekg(0,ios::end);
191
                 streampos fileInSize = fileIn.tellg();
192
193
                 ifstream fileOu(ouputFile,ios::binary);
194
                 fileOu.seekg(0,ios::end);
                 streampos fileOuSize = fileOu.tellg();
195
196
197
                 cout <<"\nKích thước file gốc: "
                                                                 <<formatFileSize(fileInSize) << endl;</pre>
198
                 cout << "Kích thước file nén: "
                                                            <<formatFileSize(fileOuSize) << endl;</pre>
199
                 double fileSizeIn = static_cast<double>(fileInSize);
200
201
                 double fileSizeOut = static_cast<double>(fileOuSize);
202
203
                 double HieuSuat = (fileSizeOut / fileSizeIn) * 100.0;
204
                 cout << "Hệ số nén file: "
                                                        << HieuSuat << " %" << endl;</pre>
205
                 cout <<"Thời gian thực thi: "
206
                                                         <<elapsed.count() <<" giây.\n" << endl;</pre>
207
208
                 input.close();
209
                 output.close();
210
211
212
213
                    cout <<"\nThất bại do không thế mở file\n"
214
215
               system("pause");
216
```

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

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

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

```
217
      //Tiến hành giải mã với file text đã được mã hóa từ một file text không chứa chữ số
219 // sử dụng hằm rlc_decoding_without_number trên từng dòng

220 void rlc_decoding_file_without_number() {

221 string inputFile;
222
          string ouputFile;
223
224
          cout <<"Nhập tên file bạn muốn giải mã: "
225
          cin >> inputFile;
          cout << "Nhập tên file bạn muốn xuất: "
cin >> ouputFile;
226
227
228
          ifstream input(inputFile):
229
230
          ofstream output(ouputFile);
231 🛱
          if(input.is_open() && output.is_open()){
232
233
               // Bắt đầu đo thời gian
234
               auto start = std::chrono::high_resolution_clock::now();
235
236
               string line;
237 占
               while(getline(input, line)){
238
239
                   string outpuLine = rlc_decoding_without_number(line);
240
                   output << outpuLine << endl;
241
242
               // Kết thúc đo thời gian
243
               auto end = std::chrono::high_resolution_clock::now();
245
246
               // Tính thời gian đã trôi qua
               std::chrono::duration(double) elapsed = end - start;
247
               249
250
               fileIn.seekg(0,ios::end);
streampos fileInSize = fileIn.tellg();
251
252
253
254
               ifstream fileOu(ouputFile,ios::binary);
255
               fileOu.seekg(0,ios::end);
256
               streampos fileOuSize = fileOu.tellg();
257
               cout <<"\nKich thước file gốc: " <<formatFileSize(fileInSize) << endl;
258
               cout <<"Kích thước file giải mã: "
259
                                                           <<formatFileSize(fileOuSize) << endl;</pre>
260
               double fileSizeIn = static_cast<double>(fileInSize);
262
               double fileSizeOut = static_cast<double>(fileOuSize);
263
264
               double HieuSuat = (fileSizeOut / fileSizeIn) * 100.0;
265
                                                << HieuSuat << " %" << endl;
<<elapsed.count() <<" giây.\n" << endl;</pre>
               cout << "Ti lệ giải nén: "
               cout <<"Thời gian thực thi: "
267
268
269
               input.close();
270
               output.close();
271
272
273
274
               cout <<"\nThất bại do không thế mở file\n"
275
276
           system("pause");
277
```

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 = 0 \times 10;
79
             string out ="";
             int i = 0;
while(i < input.length()){</pre>
80
81 = 82 | 83 = 83
                        int count = 1;
while(i +1< input.length() && input[i] == input[i+1]){
    count ++;</pre>
84
85
86 🗏
                               i++;
                               if(count == 9){
  out += dle + to_string(count) + input[i];
  count = 0;
87
88
89
90
91 🛱
89
                        if(count >= 3){
  out += dle+ to_string(count) + input[i];
92
93
                         }else{
94
                              for(int q = 0; q < count ; q++){
  out += input[i];</pre>
95
96
97
98
99
.00
             return out:
```

Hàm mã hóa RLC với văn bản có chứa chữ số

```
269
      // Tiến hành mã hóa với file text có chứa chữ số
270
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
               auto start = std::chrono::high_resolution_clock::now();
286
287
288
               string line;
               while(getline(input, line)){
289 🗀
290
                        string outpuLine = rlc_encoding_with_number(line);
291
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."
               ifstream fileIn(inputFile,ios::binary);
303
304
              fileIn.seekg(0,ios::end);
305
              streampos fileInSize = fileIn.tellg();
306
              ifstream fileOu(ouputFile,ios::binary);
307
              fileOu.seekg(0,ios::end);
streampos fileOuSize = fileOu.tellg();
308
309
310
              cout <<"\nKích thước file gốc: "
                                                     <<formatFileSize(fileInSize) << endl;</pre>
312
              cout <<"Kích thước file nén: "
                                                   <<formatFileSize(fileOuSize) << endl;</pre>
313
              double fileSizeIn = static_cast<double>(fileInSize);
314
315
              double fileSizeOut = static_cast<double>(fileOuSize);
316
317
              double HieuSuat = (fileSizeOut / fileSizeIn) * 100.0;
318
              cout << "Hệ số nén file: "
                                              << HieuSuat << " %" << endl;</pre>
319
              cout <<"Thời gian thực thi: "
                                                 <<elapsed.count() <<" giây.\n" << endl;</pre>
320
321
322
323
              input.close();
324
              output.close();
325
326
              cout <<"\nThất bại do không thế mở file\n"
327
328
329
          system("pause");
```

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

```
//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 = 0 \times 10;
107
          string out ="";
108
          int i = 0;
109 🛱
          while(i < input.length()){
110 🖨
              if(input[i] == dle){
                  int count = input[i+1] - '0';
111
112
113 🖨
                   for( int j = 0; j < count-1; j++){</pre>
114
                       out += input[i+2];
115
                   i+=2;
116
117
118
              out += input[i];
119
              i++;
120
121
          return out;
122 L
123
```

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

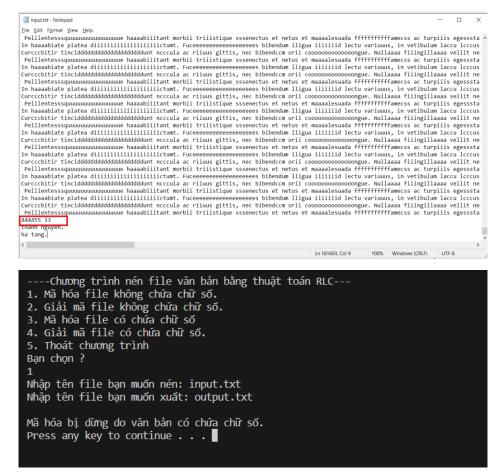
```
332 // Tiến hành giải mã với file text được mã hóa từ file text có chứa chữ số
333 // sử dụng hàm rlc_decoding_with_number trên từng dòng
334 void rlc_decoding_file_with_number() {
           string inputFile;
336
           string ouputFile;
337
           cout <<"Nhập tên file bạn muốn giải mã: "
338
339
           cin >> inputFile;
           cout << "Nhập tên file bạn muốn xuất: "
cin >> ouputFile;
340
341
342
343
           ifstream input(inputFile);
344
           ofstream output(ouputFile);
345 🖨
           if(input.is_open() && output.is_open()){
347
                // Bắt đầu đo thời gian
                auto start = std::chrono::high_resolution_clock::now();
348
349
350
                string line;
351白
                while(getline(input, line)){
352
353
                     string outpuLine = rlc_decoding_with_number(line);
354
                    output << outpuLine << endl;
355
356
357
                // Kết thúc đo thời gian
358
                auto end = std::chrono::high_resolution_clock::now();
359
                // Tính thời gian đã trôi qua
360
                std::chrono::duration(double) elapsed = end - start;
361
362
363
                cout <<"\nGiải mã thành công."
                                                          << endl;
364
                ifstream fileIn(inputFile,ios::binary);
365
                fileIn.seekg(0,ios::end);
                streampos fileInSize = fileIn.tellg();
366
367
368
                ifstream fileOu(ouputFile,ios::binary);
               fileOu.seekg(0,ios::end);
streampos fileOuSize = fileOu.tellg();
369
370
371
               cout <<"\nKich thước file gốc: "
cout <<"Kích thước file giải mã: "</pre>
372
                                                           <<formatFileSize(fileInSize) << endl;</pre>
373
                                                               <<formatFileSize(fileOuSize) << endl:</pre>
374
375
                double fileSizeIn = static_cast<double>(fileInSize);
376
                double fileSizeOut = static_cast<double>(fileOuSize);
377
378
                double HieuSuat = (fileSizeOut / fileSizeIn) * 100.0;
379
                                                    << HieuSuat << " %" << endl;
<<elapsed.count() <<" giây.\n" << endl;</pre>
                cout << "Tỉ lệ giải nén: "
380
                cout <<"Thời gian thực thi: "
381
382
383
                input.close();
384
                output.close();
385
386
387
                cout <<"\nThất bại do không thế mở file\n"
388
389
           system("pause");
391 }
```

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



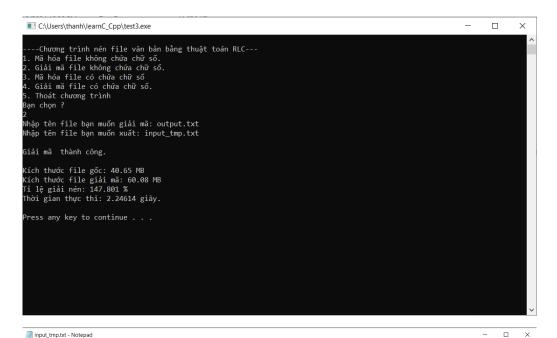
```
Chương trình nén file văn bản bằng thuật toán RLC---
1. Mã hóa file không chứa chữ số.
2. Giải mã file không chứa chữ số.
3. Mã hóa file có chữa chữ số
4. Giải mã file có chứa chữ số.
5. Thoát chương trình
Bạn chọn ?
Nhập tên file bạn muốn nén: input.txt
Nhập tên file bạn muốn xuất: output.txt
Kích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

File Edit Format View Help
Loreem 6ipum dolor 4it amet, 33cCcc6C4consectetur ad3ipicing el3it. Q4uiae ac libero vel rii3u gravida 3sceleique. 34Integeer ed riiuus eu In habiate platea d20ictumt. Fuc19es bibendum lligua 7id lectu vari3us, in vetibulum laccu l3cus 13cuet. Ut f3c3ilii lligua nec justo 20po Cur3cbitir tinci20dunt n3cula ac riiuus gittis, nec bibendccm orii c14ongue. Null4a f3ingill4a vellit nec justo congue, in fel7rentum 4sem Pe3lente3sq16ue hab3itant morbii tr3istique 3senectus et netus et m4alesuada 11famecss ac turp3is ege3stas. Sed auctor, n3isii ut ferment Loreem 6ipum dolor 4it amet, 33cCcc6C4consectetur ad3ipicing el3it. Q4uiae ac libero vel rii3u gravida 3sceleique. 34Integeer ed riiuus eu In habiate platea d20ictumt. Fuc19es bibendum lligua 7id lectu vari3us, in vetibulum laccu 13cus 13cuet. Ut f3c3ilii lligua nec justo 20po Cur3cbitir tinci20dunt n3cula ac riiuus gittis, nec bibendccm orii c14ongue. Null4a f3ingill4a vellit nec justo congue, in fe17rentum 4sem Pe3lente3sq16ue hab3itant morbii tr3istique 3senectus et netus et m4alesuada 11famecs ac turp3is ege3stas. Sed auctor, n3isii ut ferment Loreem 6ipum dolor 4it amet, 33cCcc6C4consectetur ad3ipicing el3it. Q4uiae ac libero vel rii3u gravida 3sceleique. 34Integeer ed riiuus eu In habiate platea d20ictumt. Fuc19es bibendum lligua 7id lectu vari3us, in vetibulum laccu 13cus 13cuet. Ut f3c3ilii lligua nec justo 20po Cur3cbitir tinci20dunt n3cula ac riiuus gittis, nec bibendccm orii c14ongue. Null4a f3ingill4a vellit nec justo congue, in fe17rentum 4sem 10 pe3lente3sq16ue hab3itant morbii tr3istique 3senectus et netus et m4alesuada 11famecs ac turp3is ege3stas. Sed auctor, n3isii ut ferment Loreem 6ipum dolor 4it amet, 33cCcc6C4consectetur ad3ipicing el3it. Q4uiae ac libero vel rii3u gravida 3sceleique. 34Integeer ed riiuus eu In habiate platea d20ictumt. Fuc19es bibendum lligua 7id lectu vari3us, in vetibulum laccu 13cus 13cuet. Ut f3c3ilii lligua nec justo 20po Cur3cbitir tinci20dunt n3cula ac riiuus gittis, nec bibendccm orii c14ongue In 1. Col 1 100% Windows (CRLF) LITE-8

output.txt - Notepad

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

output.txt - Notepad Ln 1, Col 1 100% Windows (CRLF) UTE-8



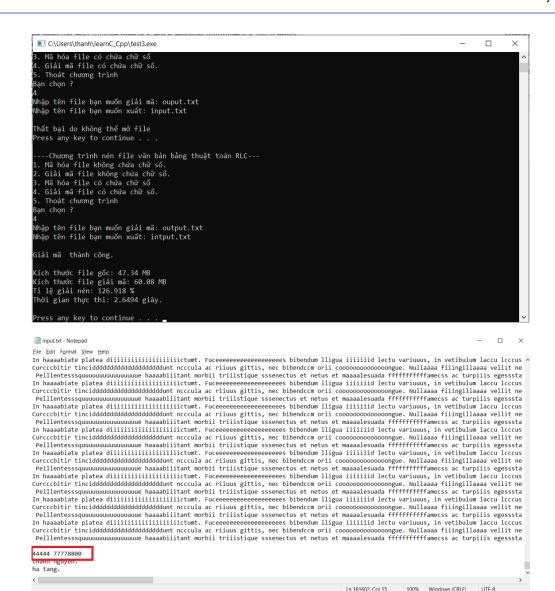
IV.3. Kiểm thử chức năng mã hóa file text có chứa chữ số

In 191603 Col 15 100% Windows (CPLE) LITE-9



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





KÉT 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.

PHŲ LŲC

Mã nguồn của chương trình : $\underline{\text{https://github.com/thanhf47/project1}}$