

# BÁO CÁO ĐỒ ÁN

## MÔN HỌC: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

### Thành viên

- Nguyễn Trần Trung – 18120625

### Những câu đã làm được

- Nén 1 tập tin chuỗi
- Nén 1 thư mục chứa nhiều tập tin chuỗi
- Nén 1 tập tin có những kiểu dữ liệu còn lại
- Nén 1 thư mục có chứa nhiều tập tin với nhiều định dạng
- Nén bằng thuật toán Huffman
- Nén bằng thuật toán Lempel – Ziv – Welch (LZW)










### Công việc của mỗi thành viên

### Sơ đồ lớp









## InStream

Class

### Fields

-  `_buffCur : int`
-  `_buffer : char*`
-  `_data : long long`
-  `_fileSize : long long`
-  `_in : ifstream`
-  `_limit :`
-  `_readSize : int`
-  `_remainBit : int`
-  `blockSize : const int`








### Methods

-  `~InStream()`
-  `addData(void* p, int pos, int data, int count) : void`
-  `get(char* data, int bitCount) : void`
-  `get(int bitCount) : int`
-  `getData(void* p, int pos, int count) : int`
-  `InStream(string filePath)`
-  `resetLim() : void`
-  `setLim( lim) : void`








## OutStream

Class

### Fields

-  `_buffCur : int`
-  `_buffer : char*`
-  `_data : long long`
-  `_out : ofstream`
-  `_remainBit : int`
-  `_writeSize : int`
-  `blockSize : const int`


### Methods

-  `~OutStream()`
-  `addData(void* p, int pos, int data, int count) : void`
-  `getData(void* p, int pos, int count) : int`
-  `OutStream(string filePath)`
-  `push(const char* data, int bitCount) : void`
-  `push(int data, int bitCount) : void`
-  `writeAll() : void`


## LZW


Class


### Fields


 maxDict : const int


### Methods


 c2s(int c) : string

 decode(string inPath, string outDirPath, string dictPath) : void

 encode(string inPath, string outPath, string dictPath) : void

 exportDictionary(Dictionary& dict, string outPath) : void

 getName(string path) : string


 importDictionary(Dictionary& dict, string inPath) : void


### Nested Types


## Dictionary


Class

### Fields


 \_deDict : vector<string>


 \_enDict : vector<vector<short>>


 \_maxSize : int

 \_size : int


### Methods


 add(int prefix, int c) : void

 Dictionary(int maxSize)

 find(int prefix, int c) : int

 isFull() : bool

 operator[](int index) : string&

 size() : int

## CompressorBase

Class

### Methods

- decode(InStream& in, string outPath) : void
- encode(string inPath, OutStream& out) : void

public

## Huffman

Class

→ CompressorBase

### Methods

- \_buildDictionary(unsigned char\*\* dictionary, Node\* node, int depth, char\* code) : void
- buildDictionary(unsigned char\*\* dictionary, Node\* huffmanTree) : void
- buildHuffmanTree(string inPath, long long& outputSize) : Node\*
- decode(InStream& in, string outPath) : void
- disposeTree(Node\* node) : void
- encode(string inPath, OutStream& out) : void
- readHuffmanTree(InStream& in, Node\*& node) : void
- setBit(char\* target, int pos, int bit) : void
- writeHuffmanTree(OutStream& out, Node\* node) : void

### Nested Types

#### Compare

Struct

##### Methods

- operator()(Node\* x, Node\* y) : bool

#### Node

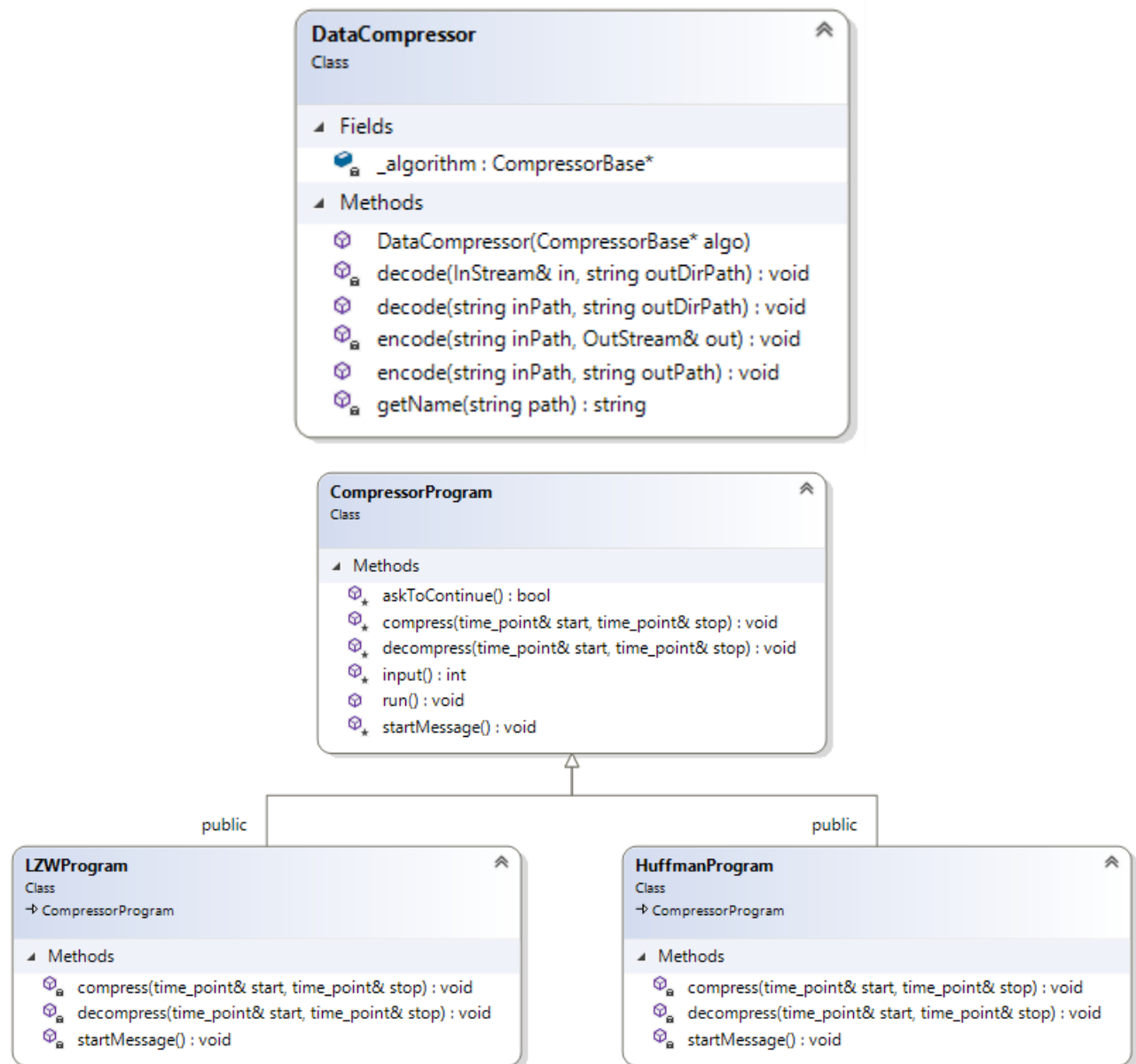
Struct

##### Fields

- chr : unsigned char
- freq : long long
- left : Node\*
- right : Node\*\*

##### Methods

- Node(long long f, char c, Node\* l, Node\* r)



## Lưu đồ thuật toán

[Link to Images](#)

## Hướng dẫn sử dụng

Nén/giải nén file và thư mục với thuật toán Huffman

```
C:\Users\nguy\ Desktop\New folder\DataCompression.exe

+-----+
| COMPRESSION PROGRAM |
| Made by: Nguyen Tran Trung |
| ID: 18120625 |
| Class: 18CTT5C |
+-----+

1. Compress file and folder using Huffman algorithm
2. Compress file using LZW algorithm
Your option: 1

-----
COMPRESS FILE AND FOLDER USING HUFFMAN ALGORITHM
1 to compress, 2 to decompress: 1
Target to compress: _
```

Ở màn hình bắt đầu, chọn 1 để vào chương trình nén/giải nén Huffman. Sẽ có 2 lựa chọn, ta chọn 1 để nén.

```
C:\Users\nguy\ Desktop\New folder\DataCompression.exe

+-----+
| COMPRESSION PROGRAM |
| Made by: Nguyen Tran Trung |
| ID: 18120625 |
| Class: 18CTT5C |
+-----+

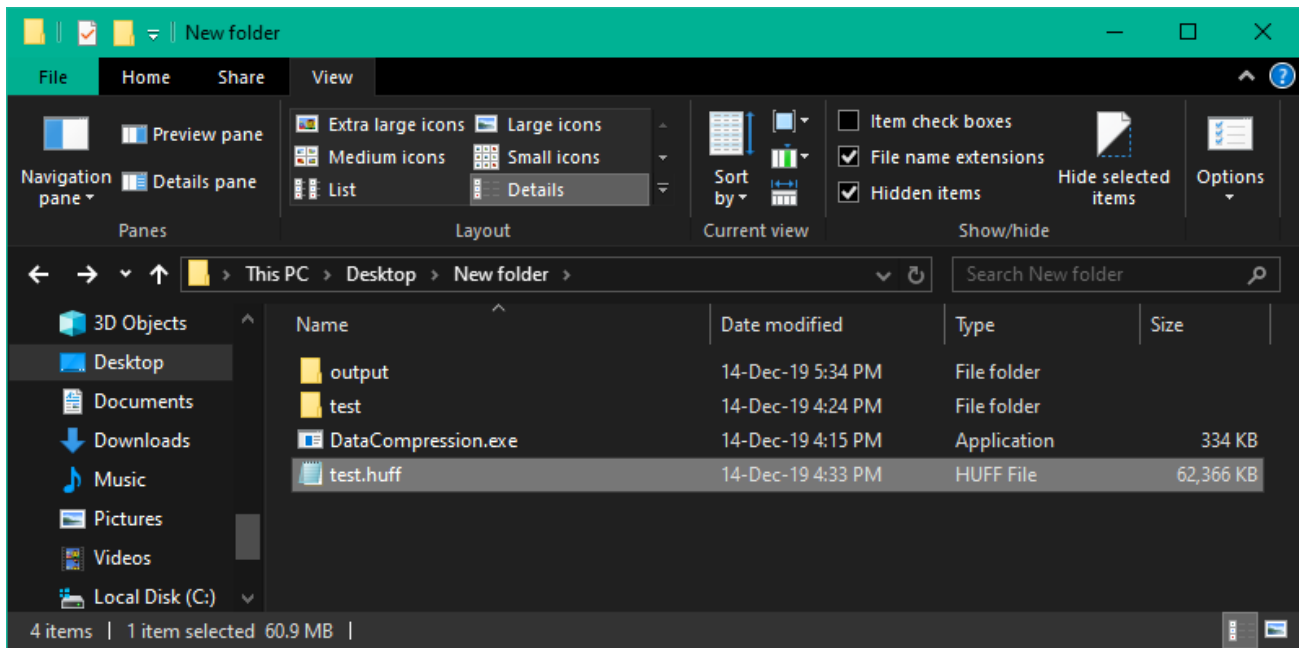
1. Compress file and folder using Huffman algorithm
2. Compress file using LZW algorithm
Your option: 1

-----
COMPRESS FILE AND FOLDER USING HUFFMAN ALGORITHM
1 to compress, 2 to decompress: 1
Target to compress: C:\Users\nguy\ Desktop\New folder\test
Output file (*.huff): C:\Users\nguy\ Desktop\New folder\test.huff
Compressing...
Processing time: 42.376s
'x' to exit, any to restart: _
```

Nhập đường dẫn file/folder cần nén sau **Target to compress**. Sau đó nhập đường dẫn của file sẽ được nén vào (có phần mở rộng là \*.huff) sau **Output file (\*.huff)**. Nhấn Enter.

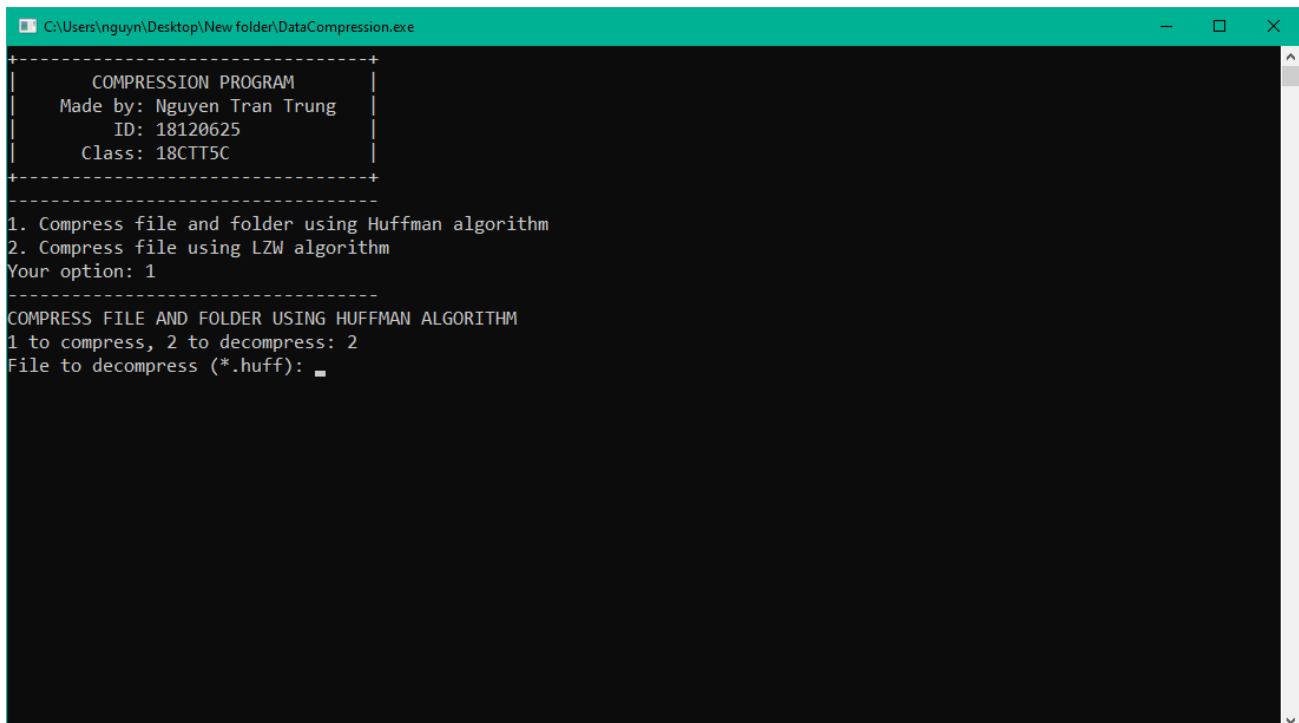
Sau khi nén xong sẽ có thông báo thời gian nén.

Ở ví dụ trên chúng ta đã nén thư mục test vào file **test.huff**. Tổng thời gian nén là **42.376s**.



'x' to exit, any to restart: nhập kí tự x để trở lại màn hình bắt đầu, nhập bất kì ký tự nào khác để thực hiện lại chương trình Huffman.

Để giải nén, ta chọn 2 ở phần lựa chọn của chương trình Huffman.



File to decompress (\*.huff): Nhập đường dẫn file \*.huff

Output directory: Nhập thư mục sẽ chứa file/folder sau khi giải nén

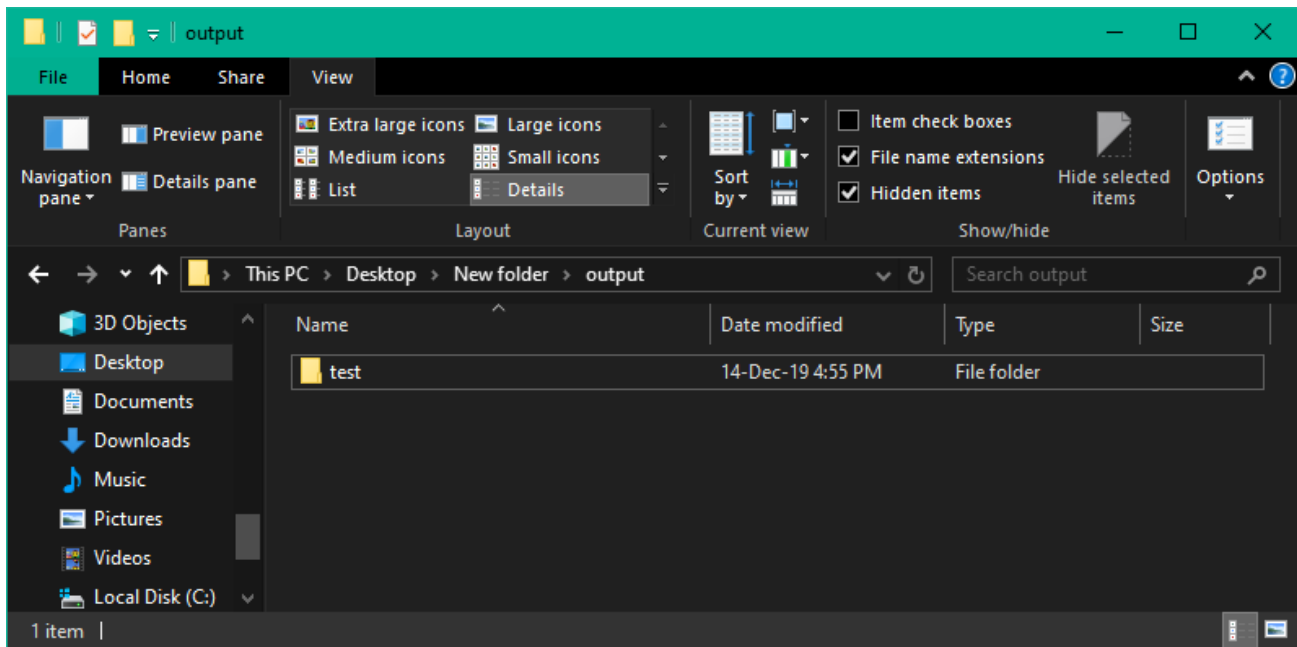
```
C:\Users\nguyn\Desktop\New folder\DataCompression.exe

COMPRESSION PROGRAM
Made by: Nguyen Tran Trung
ID: 18120625
Class: 18CTT5C

1. Compress file and folder using Huffman algorithm
2. Compress file using LZW algorithm
Your option: 1

COMPRESS FILE AND FOLDER USING HUFFMAN ALGORITHM
1 to compress, 2 to decompress: 2
File to decompress (*.huff): C:\Users\nguyn\Desktop\New folder\test.huff
Output directory: C:\Users\nguyn\Desktop\New folder\output
Decompressing...
Processing time: 21.148s
'x' to exit, any to restart: _
```

Ở ví dụ trên, ta đã giải nén file **test.huff** vào thư mục output, thời gian giải nén là **21.148s**.



Nén/giải nén file file với thuật toán LZW (Lempel – Ziv – Welch)



```
CA\Users\nguy\ Desktop\New folder\DataCompression.exe
+-----+
| COMPRESSION PROGRAM |
| Made by: Nguyen Tran Trung |
| ID: 18120625 |
| Class: 18CTT5C |
+-----+
1. Compress file and folder using Huffman algorithm
2. Compress file using LZW algorithm
Your option: 2
+-----+
COMPRESS FILE USING LZW ALGORITHM
1 to compress, 2 to decompress: 1
Target to compress:
```

Tại màn hình bắt đầu, chọn 2 để vào chương trình nén giải nén LZW. Tiếp tục chọn 1 để thực hiện nén file.

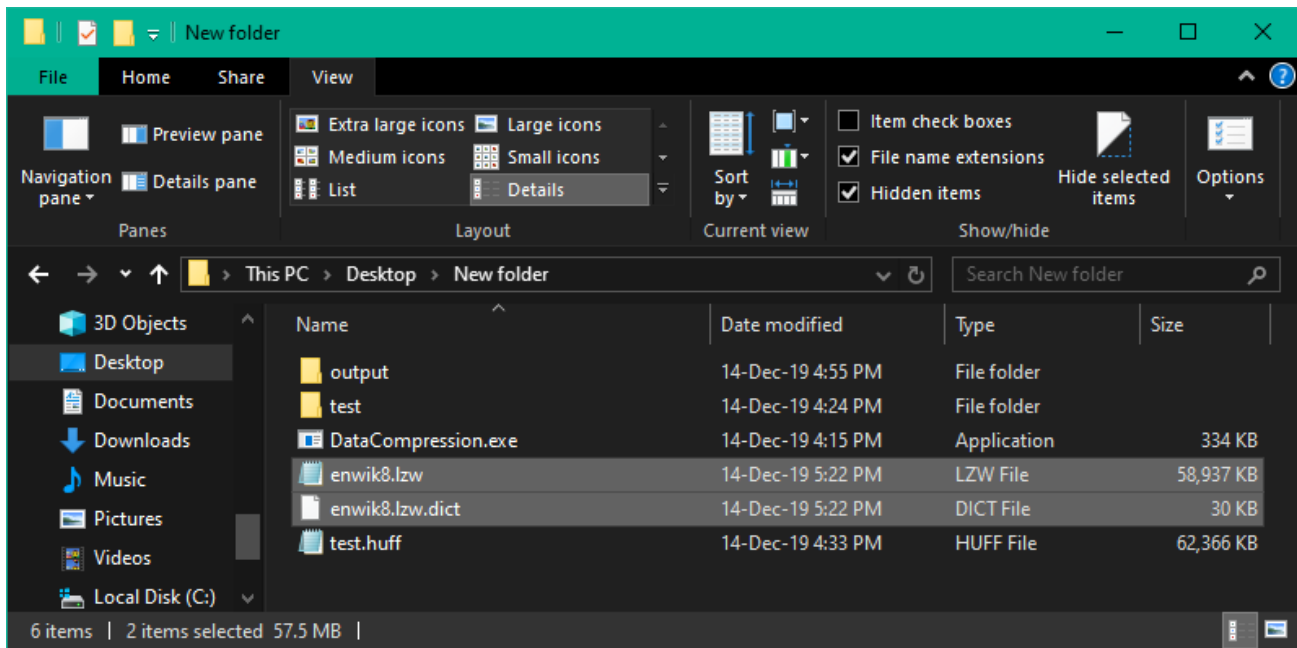
**Target to compress:** Đường dẫn file muốn nén

**Output file (\*.lzw):** Đường dẫn file muốn nén vào (phần mở rộng \*.lzw)

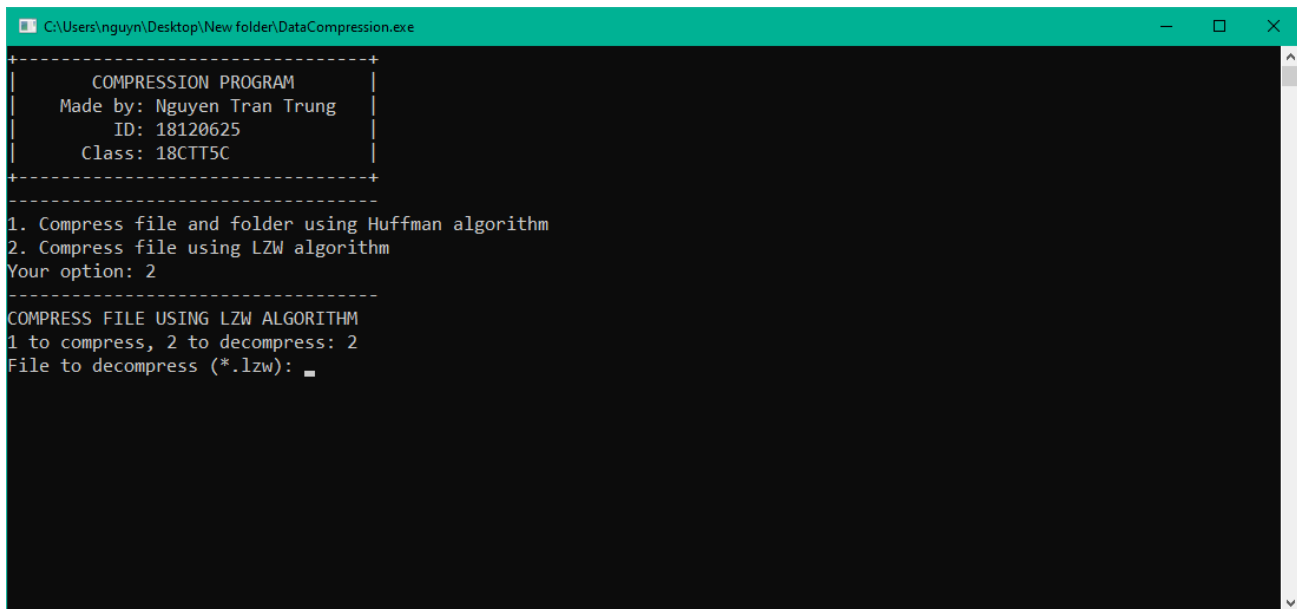
Nhấn **enter** để nén. Chương trình đồng thời sẽ tạo 1 file từ điển cùng tên với file nén với phần mở rộng là **\*.lzw.dict** ở cùng thư mục với file nén.

```
CA\Users\nguy\ Desktop\New folder\DataCompression.exe
+-----+
| COMPRESSION PROGRAM |
| Made by: Nguyen Tran Trung |
| ID: 18120625 |
| Class: 18CTT5C |
+-----+
1. Compress file and folder using Huffman algorithm
2. Compress file using LZW algorithm
Your option: 2
+-----+
COMPRESS FILE USING LZW ALGORITHM
1 to compress, 2 to decompress: 1
Target to compress: C:\Users\nguy\ Desktop\New folder\test\enwik8
Output file (*.lzw): C:\Users\nguy\ Desktop\New folder\enwik8.lzw
Compressing...
Processing time: 36.231s
'x' to exit, any to restart: _
```

Ở ví dụ trên file **enwik8** được nén vào file **enwik8.lzw** cùng với bộ từ điển giải mã là **enwik8.lzw.dict**



Để giải nén, ta chọn 2 ở phần lựa chọn của chương trình LZW.



File to decompress (\*.lzw): đường dẫn file cần giải nén (\*.lzw)

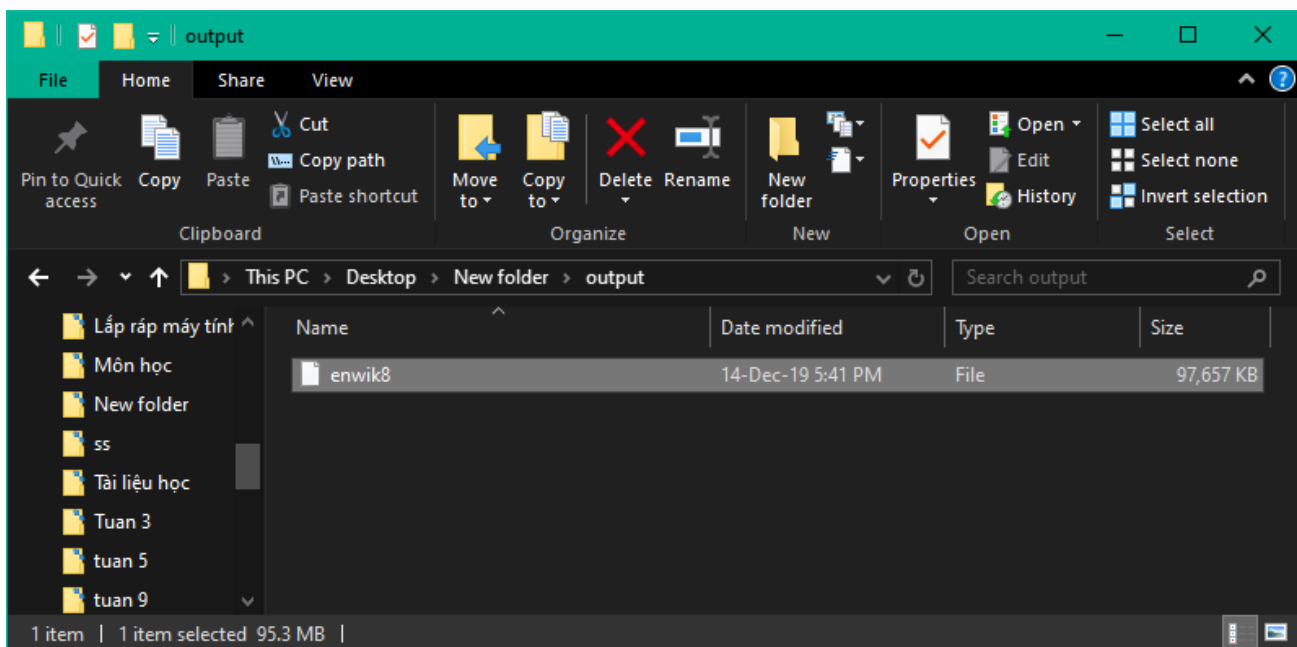
Dictionary file (\*.lzw.dict): đường dẫn file từ điển tương ứng

Output directory: đường dẫn thư mục chứa file sau khi giải nén

Nhấn **enter** để giải nén.

```
C:\Users\nguyn\Desktop\New folder\DataCompression.exe
ID: 18120625
Class: 18CTT5C
-----
1. Compress file and folder using Huffman algorithm
2. Compress file using LZW algorithm
Your option: 2
-----
COMPRESS FILE USING LZW ALGORITHM
1 to compress, 2 to decompress: 2
File to decompress (*.lzw): C:\Users\nguyn\Desktop\New folder\enwik8.lzw
Dictionary file (*.lzw.dict): C:\Users\nguyn\Desktop\New folder\enwik8.lzw.dict
Output directory: C:\Users\nguyn\Desktop\New folder\output
Decompressing...
Processing time: 29.233s
'x' to exit, any to restart:
```

Trong ví dụ trên file **enwik8.lzw** đã được giải nén vào thư mục **output** trong thời gian **29.233s**



## Video hướng dẫn sử dụng

[Link Youtube](#)