* **题目：将任意文件进行哈夫曼编码并进行压缩**
* **班级：2021211312 姓名：杨天宇 学号：2021211281** 分工： **5,6**
* **班级：2021211312 姓名：王昊钰 学号：2021211282** 分工： **1,2**
* **班级：2021211312 姓名：李志茂 学号：2021211289** 分工： **3,4**

1. 需求分析

输入输出形式：

文档中的字符均为扩充ASCII码表示。

在同级文件夹中放一个in.txt输出一个压缩后文件lock.txt

在同级文件夹中放一个lock.txt,输出一个解压文件lock.txt

程序能达到的功能

将一个文件读入，经过哈夫曼编码之后，对源文件进行压缩，输出压缩后的文件.

将一个压缩后的文件读入，解压缩后输出原文件。

输出结果

若正确压缩和解压，输出Coding success!

若无法解压该文件，输出Error:It is not file which I can understand.

1. 概要设计

解决思路：

压缩：先统计文档中不同字符出现次数作为权重，之后根据权重大小对这些字符进行哈夫曼编码，之后在文件开头输入元数据作为对压缩文件的标识，然后以一个字节为单位进行压缩并输出至目标文件中。

解压：读取文档开头的元数据检查是否可以解压，之后读取哈夫曼编码，对文件内容进行解压并输出，根据元数据中保存的最后有效输入长度处理最后一个字节。

数据结构类型：

哈夫曼树，哈夫曼树结点，用于生成哈夫曼编码

元数据结构，用于标识文件是否为程序所压缩文件，错误时的输出以及最后一个字节的数据。

字符频度表，用于统计各字符出现次数

主程序流程

压缩：初始化哈夫曼树与字符频度表，之后对文档进行哈夫曼编码。编码完成后首先向输出文件中输出元数据用于标识，之后输出压缩后的哈夫曼编码表，最后将所有内容压缩后输出到目标文件

解压：首先读取元数据确定是否能够解压，错误则输出错误提示。之后读取哈夫曼编码表，最后对压缩内容进行解压。

函数调用：

压缩：

调用initHtree和createHtree初始化哈夫曼树。

调用getHCode对文档进行哈夫曼编码

调用getlastValidBIt对最后一个字节进行特殊处理

解压：

调用initHtree和createHtree初始化哈夫曼树。

调用Hdecoding解压文件

3、详细设计

4、调试分析报告

5、用户使用说明

6、测试结果