实习报告

题目：哈夫曼码的编/译码系统

班级：电信1808 姓名：瞿博文

学号：U201813460

需求分析

利用赫夫曼编码进行通信可以大大提高信道利用率，缩短信息传输时间，降低传输成本。但是，这要求在发送端通过一个编码系统对待传数据预先编码，在接收端讲传来的数据进行译码（复原）。对于双工信道（即可以双向传输信息的信道），每端都需要一个完整的编/译码系统。

首先输入空格以及A~Z的权重，之后输入需要编码的字符串，即可得到该字符串的译码，再输入需要译文的编码，后面需要加上‘#’，即可得到该编码的译文。

测试数据

输入空格和A~Z字母的频度：

186 64 13 22 32 103 21 15 47 57 1 5 32 20 57 63 15 1 48 51 80 23 8 18 1 16 1

请输入需要编码的字符串：

THIS PROGRAM IS MY FAVORITE

该字符串编码为：

1101000101100011111100010001010011000010010101011001011101100011111110010100011111110011101011000001001001001101101010

输入需要译文的编码（以#号结束）：

1101000101100011111100010001010011000010010101011001011101100011111110010100011111110011101011000001001001001101101010#

编码译文为：

THIS PROGRAM IS MY FAVORITE

概要设计

为实现上述功能，应以树来表示。

详细设计

1.定义了储存结构

typedef struct

{

char ch

int weight;

int mother, lchild, rchild;

}DATA;

typedef struct

{

char code[30];

int cnt;

}codetype;

2.void Createtree(DATA \*hfmTree, int N)//构建哈夫曼树，传数组hfmTree和字符个数N做参数

3.void Hfmcode(DATA \*hfmTree, codetype \*codeFile, int N)//哈夫曼编码

4.void Decode(DATA \*hfmTree,char \*ToBeTran, int N)//解码过程

用户使用说明

首先输入空格以及A~Z的权重，之后输入需要编码的字符串，即可得到该字符串的译码，再输入需要译文的编码，后面需要加上‘#’，即可得到该编码的译文。

测试结果

输入空格和A~Z字母的频度：

186 64 13 22 32 103 21 15 47 57 1 5 32 20 57 63 15 1 48 51 80 23 8 18 1 16 1

请输入需要编码的字符串：

THIS PROGRAM IS MY FAVORITE

该字符串编码为：

1101000101100011111100010001010011000010010101011001011101100011111110010100011111110011101011000001001001001101101010

输入需要译文的编码（以#号结束）：

1101000101100011111100010001010011000010010101011001011101100011111110010100011111110011101011000001001001001101101010#

编码译文为：

THIS PROGRAM IS MY FAVORITE

具体代码见 .c文件