**南昌航空大学**

**21学年—22学年第 2 学期 智能医疗信息处理专业实验**

**实验二**

专业名称： 生物医学工程 实验学时： 2

学号： 19084129 姓名： 李奕澄

实验题目：二叉树的应用－哈夫曼编码的实现

实验环境： dev c++

实验目的：

1．掌握二叉树的定义；

2．掌握哈夫曼树和哈夫曼编码算法的实现。

实验内容：

实现一个哈夫曼编码系统，系统包括以下功能：

(1) 字符信息统计：读取待编码的源文件SourceFile.txt，统计出现的字符及其频率。

(2) 建立哈夫曼树：根据统计结果建立哈夫曼树。

(3) 建立哈夫曼码表：利用得到的哈夫曼树，将各字符对应的编码表保存在文件Code.txt中。

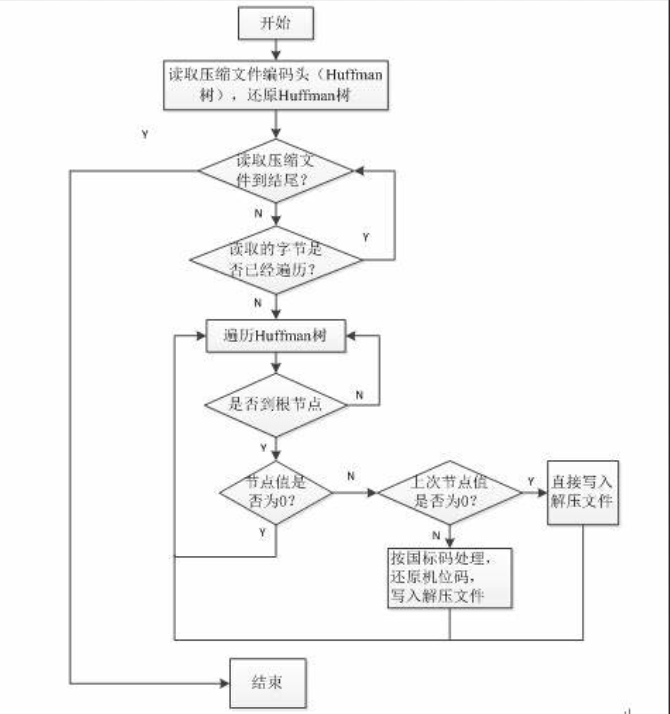
(4) 对源文件进行编码：根据哈夫曼码表，将SourceFile.txt中的字符转换成相应的编码文件ResultFile.txt。

实验要求：

(1) 程序要具在一定的健壮性，即当输入数据非法时，程序也能适当地做出反应。

(2) 程序要添加适当的注释，程序的书写要采用缩进格式。

实验流程图：



实验程序：

#define \_CRT\_SECURE\_NO\_WARNINGS

#include<iostream>

#include<cstring>

#include<cstdio>

#include<fstream>

#include<string>

using namespace std;

typedef struct {

int weight;

int parent, lchild, rchild;

}HTNode, \* HuffmanTree;

typedef char\*\* HuffmanCode;

int a[24 + 1], b[24 + 1];

char c[24 + 1];

void Select(HuffmanTree& HT, int n, int& s1, int& s2) {

for (int i = 1; i <= n; i++)

if (HT[i].parent == 0 && s1 == 0) { s1 = i; break; }

for (int i = 1; i <= n; i++)

if (HT[i].parent == 0 && HT[i].weight < HT[s1].weight) s1 = i;

for (int i = 1; i <= n; i++)

if (HT[i].parent == 0 && s2 == 0 && i != s1) { s2 = i; break; }

for (int i = 1; i <= n; i++)

if (HT[i].parent == 0 && HT[i].weight < HT[s2].weight && i != s1) s2 = i;

}

void CreateHuffmanTree(HuffmanTree& HT, int n) {

if (n <= 1) return;

int m = 2 \* n - 1;//构造一棵哈夫曼树所需要的所有节点数

HT = new HTNode[m + 1];//m+1指从第一个结点开始，第0个结点不用

for (int i = 1; i <= m; i++) {

HT[i].parent = 0;

HT[i].lchild = 0;

HT[i].rchild = 0;

}

for (int i = 1; i <= n; i++) HT[i].weight = b[i];//把字母出现的次数赋值给哈夫曼树的weight

for (int i = n + 1; i <= m; i++) {

int s1 = 0, s2 = 0;

Select(HT, i - 1, s1, s2);

HT[s1].parent = i;

HT[s2].parent = i;

HT[i].lchild = s1;

HT[i].rchild = s2;

HT[i].weight = HT[s1].weight + HT[s2].weight;

}

}

void CreateHuffmanCode(HuffmanTree& HT, HuffmanCode& HC, int n) {

HC = new char\* [n + 1];

char\* cd = new char[n];

cd[n - 1] = '\0';

for (int i = 1; i <= n; i++) {

int start = n - 1, c = i, f = HT[i].parent;

while (f) {

--start;

if (HT[f].lchild == c) cd[start] = '0';

else cd[start] = '1';

c = f;

f = HT[f].parent;

}

HC[i] = new char[n - start];

strcpy(HC[i], &cd[start]);

}

//delete cd;

}

int main() {

freopen("SourceFile.txt", "r", stdin);//打开输入文件

memset(a, 0, sizeof(a));

string str;

getline(cin, str);//接收文件的字符串并保存到str中

for (int i = 0; i < str.length(); i++)//统计每个字母出现的频率即次数

a[str[i] - 64]++;

int n = 0;

for (int i = 1, j = 1; i <= 24; i++)//统计有多少个不同的字母以及把没出现的字母去除掉

if (a[i] != 0) {

n++;

b[j] = a[i];

c[j++] = i + 64;

}

HuffmanTree HT;

HuffmanCode HC;

CreateHuffmanTree(HT, n);

CreateHuffmanCode(HT, HC, n);

freopen("Code.txt", "w", stdout);

for (int i = 1; i <= n; i++)

cout << c[i] << ':' << HC[i] << endl;

freopen("ResultFile.txt", "w", stdout);

for (int i = 0; i < str.length(); i++) {

for (int j = 1; j <= n; j++) {

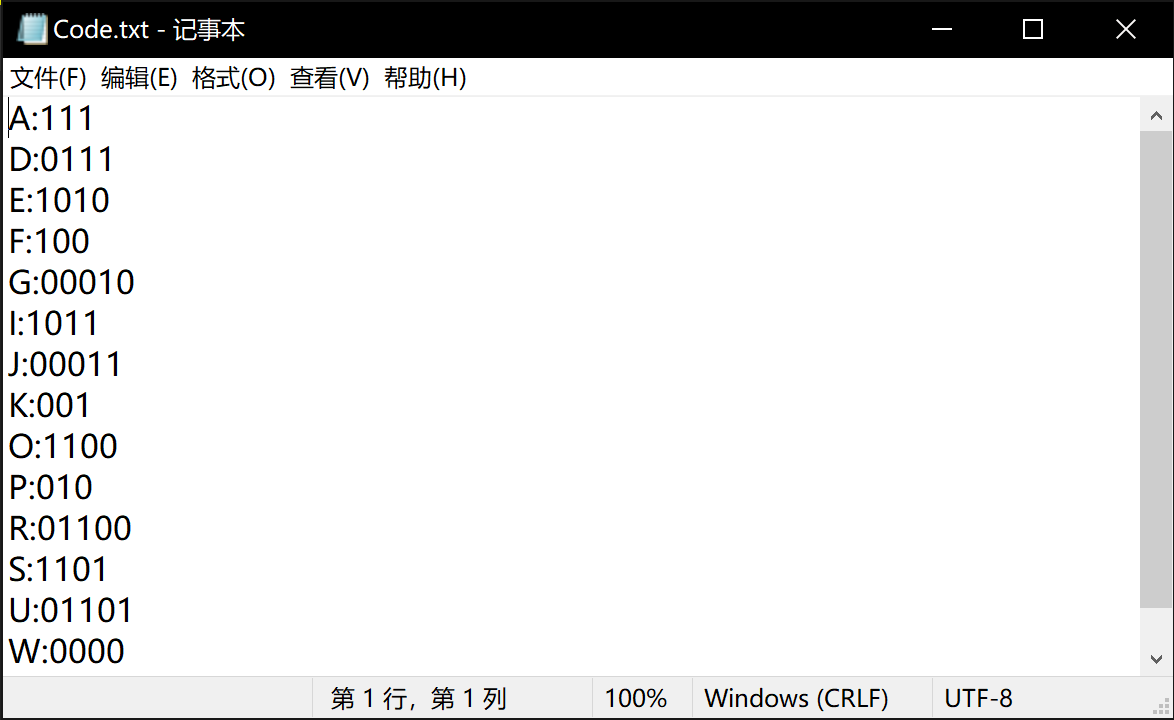
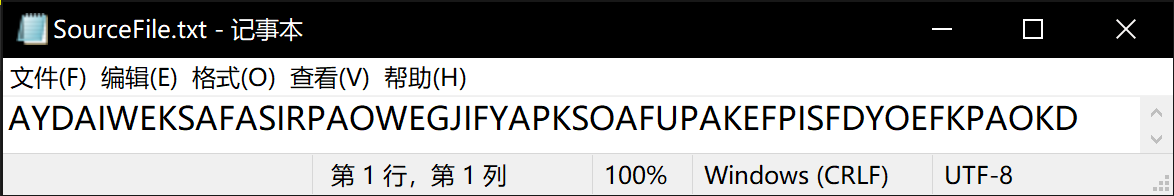
if (str[i] == c[j]) cout << HC[j];

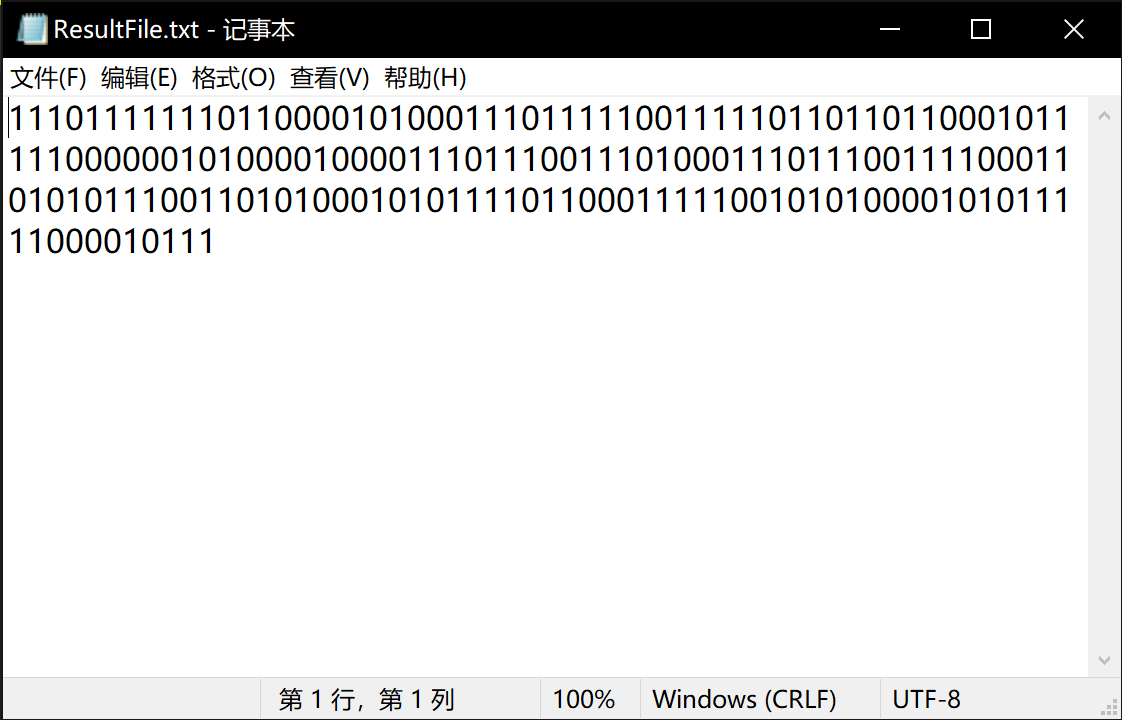
}

}

return 0;

}

**实验结果：**



心得体会：

掌握了二叉树的定义、哈夫曼树编码译码的原理与方法，并了解掌握了读写文件的方法。