代码：

#include<iostream>

#include<stdio.h>

#include<string>

#include<cstring>

#include<algorithm>

#define MAX 0x3f3f3f3f

using namespace std;

typedef struct

{

char data;

int weight;

int parent;

int lchild;

int rchild;

bool tag;

}Huffnode,\*HuffmanTree;

typedef struct

{

string \*code;

char \*data;

int num;

}HuffmanCode;

void DisplayHuffmanCode(HuffmanCode &HC,int n)

{

for(int i=0;i<n;++i)

{

cout<<HC.code[i]<<endl;

}

}

void Select(HuffmanTree &HT,int index, int &s1, int &s2)

{

int min1=MAX;

int min2=MAX;

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

{

if(HT[i].parent==0&&HT[i].tag)

{

if(HT[i].weight<min1)

{

s1=i;

min1=HT[i].weight;

}

}

}

HT[s1].tag=0;

for(int s=1;s<=index;++s)

{

if(HT[s].parent==0&&HT[s].tag)

{

if(HT[s].weight<min2)

{

s2=s;

min2=HT[s].weight;

}

}

}

HT[s2].tag=0;

}

void CreatHuffmanTree(HuffmanTree &HT,int n)

{

if(n<=1)return;

int m=2\*n-1;

HT=new Huffnode[m+1];

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

{

HT[i].lchild=0;

HT[i].rchild=0;

HT[i].parent=0;

HT[i].tag=1;

HT[i].data='\0';

}

cout<<"请依次输入每一个字和它的权重："<<endl;

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

{

scanf("%c",&HT[s].data);

getchar();

scanf("%d",&HT[s].weight);

getchar();

}

int s1,s2;

for(int z=n+1;z<=m;++z)

{

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

HT[s1].parent=z;

HT[s2].parent=z;

HT[z].lchild=s1;

HT[z].rchild=s2;

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

}

}

void CreatHuffmanCode(HuffmanTree HT, HuffmanCode &HC, int n)

{

HC.code=new string[n+1];

HC.data=new char[n+1];

HC.num=n+1;

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

{

int c=i;

int f=HT[i].parent;

string cd="";

while(f!=0)

{

if (HT[f].lchild==c)

{

cd+="0";

}

else

{

cd+="1";

}

c=f;

f=HT[f].parent;

}

reverse(cd.begin(),cd.end());

HC.code[i]=cd;

}

cout<<"当前的哈夫曼编码表为："<<endl;

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

{

HC.data[x]=HT[x].data;

cout<<x<<" | "<<HC.data[x]<<" | "<<HC.code[x]<<endl;

}

}

void Encoding(HuffmanCode HC,string str)

{

cout<<"编码结果为："<<endl;

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

{

for(int j=1;j<=HC.num;++j)

{

if(HC.data[j]==str[i])

{

cout<<HC.code[j];

break;

}

}

}

}

void Decoding(HuffmanCode HC,string str)

{

cout<<"解码结果为："<<endl;

string temp="";

int postion=0;

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

{

for(int j=1;j<=HC.num;++j)

{

int k=0;

for(;k<HC.code[j].length();++k)

{

if(HC.code[j][k]!=str[i+k])

{

break;

}

}

if(k==HC.code[j].length())

{

i+=HC.code[j].length();

cout<<HC.data[j];

break;

}

}

}

}

int main()

{

HuffmanTree HT;

HuffmanCode HC;

//构造赫夫曼树，输出各字符的赫夫曼编码

cout<<"\n请输入编码的个数："<<endl;

int n=0;

cin>>n;

getchar();

CreatHuffmanTree(HT,n);

CreatHuffmanCode(HT,HC,n);

// 编码：输入字符序列，输出对应的赫码序列。

cout<<"\n请输入要编码的字符："<<endl;

string e\_str="";

getline(cin,e\_str);

Encoding(HC,e\_str);

// 译码：输入赫夫曼码序列，输出原始字符代码。

cout<<"\n请输入要解码的字符："<<endl;

string d\_str="";

getline(cin,d\_str);

Decoding(HC,d\_str);

}

哈夫曼编码，先将英文文本构造一棵哈夫曼树，使树的左边分支标记为0，树的右边分支标记为1，每一个编码都是由根结点到该结点的分支标记组成的，即一串0和1的组合，并且每一个编码都不是其他所有编码的前缀编码。