#include <iostream>

using namespace std;

typedef struct{

char alp;

int weight;

int parent, lchild, rchild;

}HTNode, \* HuffmanTree;

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

s1 = 0; s2 = 0;

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

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

if (s1 == 0)s1 = i;

else if(HT[i].weight < HT[s1].weight)s1 = i;

}

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

if (HT[i].parent==0&&i != s1)

if (s2 == 0)s2 = i;

else if(HT[i].weight < HT[s2].weight)s2 = i;

}

return;

}

void CreatHuffmanTree(HuffmanTree& HT, int n) {

if (n <= 1)return;

int m = 2 \* n - 1;

HT = new HTNode[m + 1];

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) {

cin >>HT[i].alp>> HT[i].weight;

}

int s1 = 0, s2 = 0;

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

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;

}

}

typedef char\*\* HuffmanCode;

void CreatHuffmancode(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;

int c = i, f = HT[i].parent;

while (f != 0) {

--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\_s(HC[i],n-start,&cd[start]);

}

delete[]cd;

}

int main()

{

HuffmanTree HT;

HuffmanCode HC;

CreatHuffmanTree(HT, 7);

CreatHuffmancode(HT, HC, 7);

string temp;

cin >> temp;

for (int i = 0; i< temp.size(); ++i) {

for (int j = 1; j <= 7; ++j)

if (temp[i] == HT[j].alp)cout << HC[j]<<endl;

}

}

输入

A 3 B 12 C 7 D 4 E 2 F 8 G 11

ABC

