#include<iostream>

#include<fstream>

#include<string>

using namespace std;

class ctnode

{

public:

int x[9999], info;

char b[9999];

ctnode\* right, \* left;

};

class cnode

{

public:

int x[9999], info;

char a[9999];

cnode\* next;

ctnode\* down;

};

class tree

{

public:

ctnode\* root;

tree()

{

root = NULL;

}

~tree()

{

bye(root);

}

void insert(ctnode\* pnn)

{

if (root == NULL)

{

root = pnn;

}

else

{

if (root->left == NULL)

{

root->left = pnn;

}

else

{

root->right = pnn;

}

}

}

void bye(ctnode\* pt)

{

if (pt == NULL)

{

return;

}

bye(pt->left);

bye(pt->right);

delete pt;

}

void display\_all()

{

display(root);

}

void display(ctnode\* pt)

{

if (pt == NULL)

{

return;

}

display(pt->left);

if (pt->left == NULL && pt->right == NULL)

{

int i = 0;

while (pt->b[i] != '\0')

{

cout << pt->b[i];

i++;

}

cout << ',';

for (int i = 0; pt->x[i] == 0 || pt->x[i] == 1; i++)

{

cout << pt->x[i] << ' ';

}

cout << pt->info << ',';

cout << endl;

}

display(pt->right);

}

void remove(int v)

{

ctnode\* pb = NULL, \* pt = root;

while (pt != NULL && pt->info != v)

{

}

}

};

class CList

{

public:

cnode\* ph, \* pt;

CList()

{

ph = pt = NULL;

}

~CList()

{

cnode\* Ptrav = ph;

while (Ptrav != NULL)

{

ph = ph->next;

delete Ptrav;

Ptrav = ph;

}

}

void sorted\_attach(cnode\* pnn)

{

if (ph == NULL)

{

ph = pnn;

pt = pnn;

}

else

{

cnode\* ptrv = ph, \* pb = ph;

while (ptrv->info < pnn->info)

{

pb = ptrv;

ptrv = ptrv->next;

if (ptrv == NULL)

{

break;

}

}

if (ptrv != pb)

{

pb->next = pnn;

pnn->next = ptrv;

}

else

{

pnn->next = ph;

ph = pnn;

}

}

}

void main\_attach(cnode\* pnn)

{

if (ph == NULL)

{

ph = pt = pnn;

}

else

{

pt->next = pnn;

pt = pnn;

}

}

void disp()

{

cnode\* pt = ph;

while (pt != NULL)

{

int i = 0;

while (pt->a[i] != '\0')

{

cout << pt->a[i];

i++;

}

cout << ',';

for (int i = 0; pt->x[i] == 0 || pt->x[i] == 1; i++)

{

cout << pt->x[i] << ' ';

}

cout << ',';

cout << pt->info;

cout << endl;

pt = pt->next;

}

}

};

void decompress(CList& l, char z[], char y[], int t, int iBit, int& end)

{

int ib = 0, m = 1, f = 0, h = 0, c = 0, all = 0, i = 0, code[15];

for (i = 0; i < t - 1; i++)

{

int ex = i;

for (;;)

{

c = y[ex] & (m << ib);

if (c == 0)

{

code[h] = 0;

h++;

}

else

{

code[h] = 1;

h++;

}

ib++;

cnode\* pt = l.ph;

while (pt != NULL)

{

int s = 0, ct = 0;

for (s = 0; s < h; s++)

{

if (pt->x[s] == code[s])

{

ct++;

}

}

if (ct == h && pt->x[s] < 0)

{

h = 0;

z[f] = pt->a[0];

all = 0;

f++;

break;

}

pt = pt->next;

}

if (ib == 8)

{

ib = 0;

if (all == 0)

{

break;

}

else

{

ex++;

}

}

}

}

for (; ib < iBit; ib++)

{

c = y[i] & (m << ib);

if (c == 0)

{

code[h] = 0;

h++;

}

else

{

code[h] = 1;

h++;

}

cnode\* pt = l.ph;

while (pt != NULL)

{

int s = 0, ct = 0;

for (s = 0; s < h; s++)

{

if (pt->x[s] == code[s])

{

ct++;

}

}

if (ct == h && pt->x[s] < 0)

{

h = 0;

z[f] = pt->a[0];

f++;

break;

}

pt = pt->next;

}

}

z[f] = '\0';

end = f;

}

void read\_count(int& count)

{

fstream file;

file.open("Debug//sympols.txt", ios::in | ios::binary);

char Character;

while (file.read(&Character, 1))

{

count++;

}

}

void read\_y(char\* y)

{

fstream file;

file.open("Debug//sympols.txt", ios::in | ios::binary);

char Character;

int i = 0;

while (file.read(&Character, 1))

{

y[i] = Character;

i++;

}

}

void read\_list(CList& l)

{

fstream file;

file.open("Debug//newlist.txt", ios::in | ios::binary);

char Character;

int i = 0;

while (file.read(&Character, 1))

{

cnode\* pt = new cnode;

int i = 0;

pt->a[0] = Character;

pt->a[1] = '\0';

while (file.read(&Character, 1))

{

if (Character == '0')

{

pt->x[i] = 0;

}

if (Character == '1')

{

pt->x[i] = 1;

}

i++;

if (Character == ',')

{

break;

}

}

pt->next = NULL;

l.main\_attach(pt);

}

}

void read\_iBit(int& iBit)

{

fstream file;

file.open("Debug//iBit.txt", ios::in | ios::binary);

char Character;

int ct = 0;

while (file.read(&Character, 1))

{

ct++;

}

iBit = ct;

}

void text(char z[], int ct)

{

fstream file;

file.open("Debug//text.txt", ios::out | ios::binary);

int i = 0;

while (i < ct)

{

file << z[i];

i++;

}

}

void image(char z[], int ct)

{

fstream file;

file.open("Debug//3.bmp", ios::out | ios::binary);

int i = 0;

while (i < ct)

{

file << z[i];

i++;

}

}

void video(char z[], int ct)

{

fstream file;

file.open("Debug//kvideo.avi", ios::out | ios::binary);

int i = 0;

while (i < ct)

{

file << z[i];

i++;

}

}

void main()

{

CList l;

char\* y, z[999999];

int i = 0, iBit, ct;

read\_count(i);

y = new char[i];

read\_y(y);

read\_list(l);

read\_iBit(iBit);

decompress(l, z, y, i, iBit, ct);

//text(z, ct);

image(z, ct);

//video(z, ct);

}