#include<iostream>

#include<string.h>

#include<stdlib.h>

using namespace std;

class node

{

public:

char data[30];

char mean[100];

node \*left;

node \*right;

//int rb = 0;

//int lb = 0;

node()

{

left=NULL;

right=NULL;

}

friend class list;

};

class list

{

public:

node \*root;

node \*top;

list()

{

root=NULL;

top=NULL;

}

node\* create();

void in(node\*);

node\* del(node\*, char[]);

void update(node\*, char[]);

node\* min(node\* root)

{

node\* temp = root;

while(temp->left!=NULL)

{

temp=temp->left;

}

return temp;

}

};

void list::update(node \*temp,char d[])

{

if(temp!=NULL)

{

update(temp->left,d);

if(strcmp(temp->data,d)==0)

{

strcpy(temp->data,d);

strcpy(temp->mean,d);

}

update(temp->right,d);

}

}

node\* list::del(node\* root, char data[])

{

int n = strcmp(data,root->data);

if(root==NULL)

return root;

if (n<0)

root->left = del(root->left, data);

else if (n>0)

root->right = del(root->right, data);

else

{

if (root->left == NULL)

{

node \*temp = root->right;

delete(root);

return temp;

}

else if (root->right == NULL)

{

node \*temp = root->left;

delete(root);

return temp;

}

node\* temp = min(root->right);

strcpy(root->data,temp->data);

strcpy(root->mean,temp->mean);

root->right = del(root->right, temp->data);

}

return root;

}

void list::in(node\* temp)

{

if(temp!=NULL)

{

in(temp->left);

cout<<"\n Word: "<<temp->data<<"\nMeaning: "<<temp->mean<<"\n";

in(temp->right);

}

}

node\* list::create()

{

char word[30];

char m[100];

int n;

cout<<"\nEnter the word\n";

cin>>word;

cout<<"\nEnter the Meaning\n";

cin.ignore();

cin.getline(m,100);

node \*temp = new node();

strcpy(temp->data,word);

strcpy(temp->mean,m);

temp->left= NULL;

temp->right= NULL;

while(1)

{

if(root==NULL)

{

root = temp;

top = root;

break ;

}

else

{

n = strcmp(word,root->data);

if(n>0)

{

if(root->right==NULL)

{

root->right = temp;

break;

}

else

{

root = root->right;

}

}

else

{

if(root->left==NULL)

{

root->left = temp;

break;

}

else

{

root = root->left;

}

}

}

}

return top;

}

int main()

{

list l;

int ch;

char data[30];

char da[30];

while(1)

{

cout<<"\n1.Insert\n";

cout<<"\n2.Traverse\n";

cout<<"\n3.Delete\n";

cout<<"\n4.Update\n";

cout<<"\n5.Exit\n";

cout<<"\nEnter Your Choice\n";

cin>>ch;

switch(ch)

{

case 1:

l.root = l.create();

break;

case 2:

l.in(l.root);

break;

case 3:

cout<<"\nEnter the word you want to delete\n";

cin>>data;

l.root=l.del(l.root,data);

break;

case 4:

cout<<"\nEnter the word you want to update\n";

cin>>da;

l.update(l.root,data);

break;

case 5:

exit(1);

break;

default :

cout<<"\nWrong Choice\n";

}

}

return 0;

}