#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "bst.h"

char name[20];

int main(){

Tree \*bst;

char \*gchild[4];

char \*str[12] = {"Kumar","Anusha","Ram","Charan","Mohan","Karthika","Chitra","Lakshmi","Abishek","Swetha","Tarun","Sanjana"};

for(int i=0;i<12;i++)

bst = insert(str[i],bst);

int choice;

printf("BST OPERATIONS\n1: Display in Alphabetical order\n2: Find Grandparent\n3: Find Grandchildren\n4: Find Siblings\n5: Delete and display\n");

printf("\nEnter Your choice: ");

scanf("%d", &choice);

do{

switch(choice){

case 1: printf("-----------------------------------------------------------------\n");

printf("ALPHABETICAL ORDER\n");

inorder(bst);

printf("----------------------------------------------------------------\n");

break;

case 2: printf("-----------------------------------------------------------------\n");

printf("Enter the person whose grandparent you need to find: ");

scanf("%s", name);

level = 0;

findGrandparent(name,bst);

printf("-----------------------------------------------------------------\n");

break;

case 3: printf("-----------------------------------------------------------------\n");

printf("Enter the person whose grandchildren you need to find: ");

scanf("%s", name);

int ind = findGrandChildren(name,bst,gchild);

if(ind != 0){

printf("Grand Children of %s: ",name);

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

printf("%s ",gchild[i]);

printf("\n");

}

else{

printf("No grandchildern");

}

printf("-----------------------------------------------------------------\n");

break;

case 4: printf("-----------------------------------------------------------------\n");

printf("Enter the person whose Siblings you need to find: ");

scanf("%s", name);

level = 0;

findSibling(name,bst);

printf("-----------------------------------------------------------------\n");

break;

case 5: printf("-----------------------------------------------------------------\n");

printf("Enter the person whom you want to delete: ");

scanf("%s", name);

bst = delete(name,bst);

printf("Resultant Tree:\n");

inorder(bst);

printf("-----------------------------------------------------------------\n");

break;

}

printf("BST OPERATIONS\n1: Display in Alphabetical order\n2: Find Grandparent\n3: Find Grandchildren\n4: Find Siblings\n5: Delete and display\n");

printf("\nEnter Your choice: ");

scanf("%d", &choice);

}while(choice != -1);

}

**----------------------------------------------------------------------bst.h**

struct SearchTree{

char element[20];

struct SearchTree \*left,\*right;

};

typedef struct SearchTree Tree;

void inorder(Tree \*t){

if(t != NULL){

inorder(t->left);

printf("%s\n",t->element);

inorder(t->right);

}

}

Tree \*findmin(Tree \*t){

if(t==NULL)

return NULL;

else if(t->left==NULL)

return t;

else

return findmin(t->left);

}

Tree \*find(char x[],Tree \*t){

if (t==NULL)

return NULL;

if (strcmp(x,t->element)<0)

return find(x, t->left);

else if (strcmp(x,t->element)>0)

return find(x,t->right);

else

return t;

}

int level = 0;

int findLevel(char x[],Tree \*t){

if (t==NULL)

return -1;

if (strcmp(x,t->element)<0){

level++;

return findLevel(x, t->left);

}

else if (strcmp(x,t->element)>0){

level++;

return findLevel(x,t->right);

}

else{

return level;

}

}

void findGrandparent(char name[],Tree \*t){

int level = findLevel(name,t);

if(level == 0 || level == 1){

printf("No grandparent\n");

return;

}

//printf("l = %d\n",level);

int i=0;

while(i != (level - 2)){

if(strcmp(name,t->element)<0){

i++;

//printf("i=%d\n",i);

t = t->left;

}

else if(strcmp(name,t->element)>0){

i++;

//printf("i=%d\n",i);

t = t->right;

}

}

printf("The grandparent of %s is %s\n",name,t->element);

}

void findSibling(char name[],Tree \*t){

int level = findLevel(name,t);

if(level == 0){

printf("No Sibling\n");

return;

}

int i = 0;

while(i != (level - 1)){

if(strcmp(name,t->element)<0){

i++;

t = t->left;

}

else if(strcmp(name,t->element)>0){

i++;

t = t->right;

}

}

if(t->left->element==NULL || t->right->element==NULL){

printf("No Sibling\n");

return;

}

(strcmp(name,t->left->element)==0)?printf("The sibling of %s is %s\n",name,t->right->element):printf("The sibling of %s is %s\n",name,t->left->element);

}

int findGrandChildren(char name[],Tree \*t,char \*gchild[]){

Tree \*tmp = t;

tmp = find(name,tmp);

int ind = 0;

if(tmp->left != NULL){

if(tmp->left->left != NULL){

gchild[ind] = (char \*)malloc(strlen(tmp->left->left->element)\*sizeof(char));

strcpy(gchild[ind++],tmp->left->left->element);

}

if(tmp->left->right != NULL){

gchild[ind] = (char \*)malloc(strlen(tmp->left->right->element)\*sizeof(char));

strcpy(gchild[ind++],tmp->left->right->element);

}

}

if(tmp->right != NULL){

if(tmp->right->left != NULL){

gchild[ind] = (char \*)malloc(strlen(tmp->right->left->element)\*sizeof(char));

strcpy(gchild[ind++],tmp->right->left->element);

}

if(tmp->right->right != NULL){

gchild[ind] = (char \*)malloc(strlen(tmp->right->right->element)\*sizeof(char));

strcpy(gchild[ind++],tmp->right->right->element);

}

}

return ind;

}

Tree \*insert(char x[], Tree \*t){

if(t==NULL){

t=(Tree \*)malloc(sizeof(Tree));

if(t==NULL)

printf("Out of Space!!");

else{

strcpy(t->element,x);

t->left=t->right=NULL;

}

}

else if(strcmp(x,t->element)<0)

t->left=insert(x,t->left);

else if(strcmp(x,t->element)>0)

t->right=insert(x,t->right);

return t;

}

Tree \*delete(char x[], Tree \*t){

Tree \*tmpcell;

if(t==NULL)

printf("Element not found");

else if(strcmp(x,t->element)<0)

t->left=delete(x,t->left);

else if(strcmp(x,t->element)>0)

t->right=delete(x,t->right);

else if(t->left && t->right){

tmpcell=findmin(t->right);

strcpy(t->element,tmpcell->element);

t->right=delete(t->element,t->right);

}

else{

tmpcell=t;

if(t->left==NULL)

t=t->right;

else if(t->right==NULL)

t=t->left;

free(tmpcell);

}

return t;

}

**----------------------------------------------------------------------OUTPUT**

BST OPERATIONS

1: Display in Alphabetical order

2: Find Grandparent

3: Find Grandchildren

4: Find Siblings

5: Delete and display

Enter Your choice: 1

-----------------------------------------------------------------

ALPHABETICAL ORDER

Abishek

Anusha

Charan

Chitra

Karthika

Kumar

Lakshmi

Mohan

Ram

Sanjana

Swetha

Tarun

----------------------------------------------------------------

BST OPERATIONS

1: Display in Alphabetical order

2: Find Grandparent

3: Find Grandchildren

4: Find Siblings

5: Delete and display

Enter Your choice: 2

-----------------------------------------------------------------

Enter the person whose grandparent you need to find: Lakshmi

The grandparent of Lakshmi is Ram

-----------------------------------------------------------------

BST OPERATIONS

1: Display in Alphabetical order

2: Find Grandparent

3: Find Grandchildren

4: Find Siblings

5: Delete and display

Enter Your choice: 2

-----------------------------------------------------------------

Enter the person whose grandparent you need to find: Karthika

The grandparent of Karthika is Anusha

-----------------------------------------------------------------

BST OPERATIONS

1: Display in Alphabetical order

2: Find Grandparent

3: Find Grandchildren

4: Find Siblings

5: Delete and display

Enter Your choice: 3

-----------------------------------------------------------------

Enter the person whose grandchildren you need to find: Charan

Grand Children of Charan: Chitra

-----------------------------------------------------------------

BST OPERATIONS

1: Display in Alphabetical order

2: Find Grandparent

3: Find Grandchildren

4: Find Siblings

5: Delete and display

Enter Your choice: 4

-----------------------------------------------------------------

Enter the person whose Siblings you need to find: Swetha

The sibling of Swetha is Mohan

-----------------------------------------------------------------

BST OPERATIONS

1: Display in Alphabetical order

2: Find Grandparent

3: Find Grandchildren

4: Find Siblings

5: Delete and display

Enter Your choice: 4

-----------------------------------------------------------------

Enter the person whose Siblings you need to find: Chitra

No Sibling

-----------------------------------------------------------------

BST OPERATIONS

1: Display in Alphabetical order

2: Find Grandparent

3: Find Grandchildren

4: Find Siblings

5: Delete and display

Enter Your choice: 5

-----------------------------------------------------------------

Enter the person whom you want to delete: Ram

Resultant Tree:

Abishek

Anusha

Charan

Chitra

Karthika

Kumar

Lakshmi

Mohan

Sanjana

Swetha

Tarun

-----------------------------------------------------------------

BST OPERATIONS

1: Display in Alphabetical order

2: Find Grandparent

3: Find Grandchildren

4: Find Siblings

5: Delete and display

Enter Your choice: -1