**SSN COLLEGE OF ENGINEERING**

**DEPARTMENT OF CSE**

**ASSIGNMENT 8**

**NAME:S.MADHUMITHA**

**ROLL NO:185001086**

**CODE:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct tree{

char n[20];

struct tree\* left;

struct tree\* right;

}tree;

tree\* root;

tree\* insert(char name[],tree\* t){ *//inserting nodes*

if(t==NULL){

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

if(t==NULL)printf("error");

else{

strcpy(t->n,name);

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

}}

else if(strcmp(name,t->n)<1){

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

}

else if(strcmp(name,t->n)>1){

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

}

return t;

}

tree\* grandparent(char name[],tree\* t){  *//finding grandparent*

int f=0;

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

if(strcmp(t->left->left->n,name)==0){f=1;return t;}

}

else if(t->left->right!=NULL){

if(strcmp(t->left->right->n,name)==0){f=1;return t;}

}

else if(t->right->left!=NULL){

if(strcmp(t->right->left->n,name)==0){f=1;return t;}

}

else if(t->right->right!=NULL){

if(strcmp(t->right->right->n,name)==0){f=1;return t;}

}

else return NULL;

if(strcmp(t->n,name)>1){

t=grandparent(name,t->left);}

else if(strcmp(t->n,name)<1){

t=grandparent(name,t->right);}

}

tree\* findnode(char name[],tree\* t){ *//locating a node*

if(!strcmp(name,t->n))return t;

else if(strcmp(name,t->n)<1){

findnode(name,t->left);}

else if(strcmp(name,t->n)>1){

findnode(name,t->right);

}

}

void grandchildren(char name[],tree\* tr){ *//finding grandchildren*

tree\* t=findnode(name,tr); printf("found");

printf("grandchildren:\n");

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

printf("%s",t->left->right->n);

printf("%s",t->right->left->n);

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

return;

}

void inorder(tree\* t){ *//inorder traversal*

if(t!=NULL){

inorder(t->left);

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

inorder(t->right);}

}

tree\* sibling(char name[],tree\* t){ *//find sibling node*

if(!strcmp(t->right->n,name)){

if(t->left)

return t->left;

}

else if(!strcmp(t->left->n,name)){

if(t->right)

return t->right;

}

else if(strcmp(t->n,name)<1){

t=sibling(name,t->left);

}

else if(strcmp(t->n,name)>1){

t=sibling(name,t->right);

}

return t;

}

tree\* findmin(tree\* t){

if(t==NULL)return NULL;

else if(t->left==NULL)return t;

else return findmin(t->left);

}

tree\* delete(char name[],tree\* tr){ *//deleting a node*

tree\* tmp=(tree\*)malloc(sizeof(tree));

if(strcmp(name,tr->n)<1)tr=delete(name,tr->left);

else if(strcmp(name,tr->n)>1)tr=delete(name,tr->right);

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

tmp=findmin(tr);

strcpy(tr->n,tmp->n);

tr->right=delete(tr->n,tr->right);

}

else {

tmp=tr;

if(tr->left==NULL)tr=tr->right;

else if(tr->right==NULL)tr=tr->left;

}

return tr;

}

int main(){

tree \*temp;

temp=(tree\*)malloc(sizeof(struct tree));

int choice;char name[20],name1[20],c='y';

root=(tree\*)malloc(sizeof(struct tree));root=NULL;

while(c=='y'||c=='Y'){

printf("name");scanf("%s",name1);

root=insert(name1,root);

printf("add more y/n");scanf(" %c",&c);

}//inorder(root);

c='y';

while(c=='y'||c=='Y'){

printf("menu");

printf("1.grandparent of a node\n2.grandchildren of a node\n3.sibling of a node\n4.Delete a node\n");

printf("choice");scanf("%d",&choice);

switch(choice){

case 1:

printf("name");scanf("%s",name);

temp=grandparent(name,root);if(temp!=NULL)printf("%s is grandparent",temp->n);else printf("no grandparent");

break;

case 2:

printf("name");scanf("%s",name);

grandchildren(name,root);

break;

case 3:

printf("name");scanf("%s",name);

temp=sibling(name,root);

if(temp)printf("%s is sibling ",temp->n);

else printf("no sibling");

break;

case 4:

printf("name to delete");

scanf("%s",name);

temp=delete(name,root);

break;

}printf("\nenter y/n");scanf(" %c",&c);

}

Printf(“\ninorder traversal\n”); inorder(root);

}

**OUTPUT:**

namekumar

add more y/ny

nameanusha

add more y/ny

namechr ram

add more y/ny

namecharan

y

add more y/nnamek mohan

add more y/ny

namekarthika

add more y/ny

namechitra

add more y/nu y

namelakshmi

add more y/ny

nameabishek

add more y/ny

nameswetha

add more y/ny

nametarun

add more y/ny

namesanjana

add more y/nn

menu1.grandparent of a node

2.grandchildren of a node

3.sibling of a node

4.Delete a node

choice1

namelakshmi

ram

continue y/ny

choice1

namekarthika

anusha

continue y/ny

choice2

namecharan

chitra

continue y/ny

choice3

nameswetha

mohan

continue y/ny

choice3

namechitra

null

continue y/ny

choice4

nameram

continue y/nn

inorder traversal

abishek anusha charan chitra karthika kumar mohan lakshmi sanjana swetha tarun