Roll NO:27

NAME : Mali Anjali Prakash

DIV: FYMCA-A

Assignment no -6

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct Node

{

    char name[20];

    char phone[20];

    struct Node \* left;

    struct Node \*right;

}Node;

Node \*created(char name[],char phone[])

{

    Node \*newNode = (Node\*)malloc(sizeof(Node));

    strcpy(newNode->name,name);

    strcpy(newNode->phone,phone);

    newNode->left=NULL;

    newNode->right=NULL;

    return newNode;

}

Node\* insert(Node\* root,char name[],char phone[])

{

    if(root==NULL)

    return created(name,phone);

   if(strcmp(name,root->name)<0)

   {

    root->left=insert(root->left,name,phone);

   }

   else{

     root->right=insert(root->right,name,phone);

   }

   return root;

}

Node \*minimun(Node \* root)

{

    while(root&&root->left!=NULL)

    {

        root=root->left;

    }

    return root;

}

Node \* delete(Node\* root,char name[])

{

    if(root==NULL)

    {

        return root;

    }

       if(strcmp(name,root->name)<0)

       {

        root->left=delete(root->left,name);

       }

       else if(strcmp(name,root->name)>0)

       {

        root->right=delete(root->right,name);

       }

       else{

           if(root->left==NULL)

           {

            Node \* temp=root->right;

            free(root);

            return temp;

           }

           else if( root->right==NULL);

           {

             Node \*temp=root->left;

            free(root);

            return temp;

           }

       Node \*temp=minimun(root->right);

       strcpy(root->name,temp->name);

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

       root->right=delete(root->right,temp->name);

       }

       return root;

}

Node \*search(Node \* root, char name[])

{

    if(root==NULL||strcmp(root->name,name)==0)

    {

        return root;

    }

    if(strcmp(name,root->name)<0)

    {

        return search(root->left,name);

    }

    else

    {

        return search(root->right, name);

    }

}

void display(Node \*root)

{

    if(root==NULL)

    {

      return;

    }

    display(root->left);

    printf("the name is %s,phone is %s \n",root->name,root->phone);

    display(root->right);

}

  void main()

  {

    Node \*root=NULL;

    int ch;

    char name[20],phone[20];

    do

    {

      printf("1.insert\n 2.delete\n 3.serach\n4.dispaly\n 5.exit\n");

      printf("enter the choice \n");

       scanf("%d",&ch);

       switch (ch)

       {

       case 1:

                          printf("enter namen");

       scanf("%s",name);

        printf("enter phone");

        scanf("%s",phone);

         root=insert(root,name,phone);

        break;

        case 2: printf("enter name/n");

       scanf("%s",name);

       root= delete(root,name);

        break;

        case 3:printf("enter name/n");

       scanf("%s",name);

       Node\* result=search(root,name);

         if(result)

         {

            printf("the name is found %s, and phone number is %s",result->name,result->phone);

         }

         else{

            printf("Not found");

         }

        break;

        case 4:

                  display(root);

        break;

        case 5:

           printf("exiting");

        break;

       default:printf("not");

        break;

       }

    } while (ch!=5);

  }





