```
//playerIf.h
struct player{
	char name[20];
	struct player *left,*right;
};

struct player* insert(struct player *t, char n[20]);
struct player* search(struct player *t, char n[20]);
struct player* delete(struct player *t,char n[20]);
struct player* findmin(struct player *t);
void display(struct player *t,int depth);
struct player* search(struct player*t, char n[20]);
void inorder(struct player *t);
void postorder(struct player *t);
void preorder(struct player *t);

//playerImpl.h
struct player* findMin(struct player *t){
	if(t==NULL)
		return NULL;
	while(t->left!=NULL)
		return findMin(t->left);
}

struct player* insert(struct player *t, char n[20]){
	if(t==NULL){
		t=(struct player*)malloc(sizeof(struct player));
		strcpy(t->name,n);
		t->left=t->right=0;
	}
	else if(strcmp(n,t->name)<0)
```

```
                t->left=insert(t->left,n);
        else if(strcmp(n,t->name)>0)
                t->right=insert(t->right,n);
        return t;
}

struct player* delete(struct player*t,char n[20]){
        struct player *temp;
        if(t==NULL)
                return NULL;

        else if(strcmp(n,t->name)<0)
                t->left=delete(t->left,n);

        else if(strcmp(n,t->name)>0)
                t->right=delete(t->right,n);

        else if(t->left&&t->right){
                temp=findMin(t->right);
                strcpy(t->name,temp->name);
                t->right=delete(t->right,t->name);
                free(temp);
        }

   else{
                temp=t;
                if(t->right==NULL)
                        t=t->left;
      else if(t->left==NULL)
                        t=t->right;
                free(temp);
```

```
        }
        return t;
}

struct player* findmin(struct player *t)
{
        while(t->left!=NULL) //recursive method
                t=t->left;
        return t;
}

void display(struct player *t,int depth){ //preorder-PLR
        int i;
        for(i=0;i<depth;i++)
                printf("\t");
        printf("%s",t->name);
        printf("\n");
   if(t->left!=NULL)
      display(t->left,depth+1);
   if(t->right!=NULL)
      display(t->right,depth+1);
}

void inorder(struct player *t){ //LPR
        if(t->left!=NULL)
        inorder(t->left);
        //printing parent
        printf("  %s",t->name);
        if(t->right!=NULL)
        inorder(t->right);
}
```

```
void postorder(struct player *t){ //LRP
	if(t->left!=NULL)
	inorder(t->left);
	if(t->right!=NULL)
	postorder(t->right);
	//printing parent
	printf("  %s",t->name);
}

void preorder(struct player *t){ //PLR
	//printing parent
	printf("  %s",t->name);
	if(t->left!=NULL)
	inorder(t->left);
	if(t->right!=NULL)
	postorder(t->right);
}

struct player* search(struct player* t, char n[]){
	if (t==NULL)
		return NULL;
	if(strcmp(t->name,n)==0)
		return t;
	if(strcmp(t->name,n)>0)
		return search(t->left,n);
	if(strcmp(t->name,n)<0)
		return search(t->right,n);
}
```

```
//playerAppl.c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include "playerIf.h"
#include "playerImpl.h"

void main(){
        struct player *t=NULL;
        int i;
        char n[20];
        for(i=0;i<5;i++){
                printf("Name: ");
                scanf(" %[^\n]",n);
                t=insert(t,n);
        }
        display(t,0);

        char na[20];
        //Deleting
        printf("\nDelete:\tEnter name to delete: ");
        scanf(" %[^\n]",na);
        printf("After deleting %s:\n", na);
        t=delete(t,na);
        printf("\n");
        display(t,0);
        //Searching
   printf("\nSearch:\tEnter name: ");
        scanf(" %[^\n]",na);
        struct player *found=search(t,na);
        if(found==NULL)
```

```
        printf("Not found");
    else
        printf("%s is found",found->name);

            printf("\nINORDER:\n");
            inorder(t);
            printf("\nPREORDER:\n");
            preorder(t);
            printf("\nPOSTORDER:\n");
            postorder(t);
}
/*
OUTPUT:
Name: Swetha
Name: Tarun
Name: Sush
Name: Surya
Name: Sowmya
Swetha
        Sush
                Surya
                        Sowmya
        Tarun

Delete: Enter name to delete: Sush
After deleting Sush:

Swetha
        Surya
                Sowmya
        Tarun
```

Search: Enter name: Surya

Surya is found

INORDER:

Sowmya Surya Swetha Tarun

PREORDER:

Swetha Sowmya Surya Tarun

POSTORDER:

Sowmya Surya Tarun Swetha

*/