```
#include<stdio.h>
#include<stdlib.h>

struct data{
    int angka;
    data *left;
    data *right;
}*root = NULL;

void insertNode(data **temp,int angka){

    if(*temp==NULL){
        *temp = (data*)malloc(sizeof(data));
        (*temp)->angka = angka;
        (*temp)->left = NULL;
        (*temp)->right = NULL;
    }
    else if(angka > (*temp)->angka) insertNode(&(*temp)->right,angka);
    else if(angka < (*temp)->angka) insertNode(&(*temp)->left,angka);
}


void preorder(data **temp){
    if(*temp!=NULL){
        printf("%d ",(*temp)->angka);
        preorder(&(*temp)->left);
        preorder(&(*temp)->right);
    }
}

void inorder(data **temp){
    if(*temp!=NULL){
        inorder(&(*temp)->left);
        printf("%d ",(*temp)->angka);
        inorder(&(*temp)->right);
    }
}

void postorder(data **temp){
    if(*temp!=NULL){
        postorder(&(*temp)->left);
        postorder(&(*temp)->right);
        printf("%d ",(*temp)->angka);
    }
}


data **getAnakKiriPalingKanan(data**temp){
    if((*temp)->right!=NULL){
        getAnakKiriPalingKanan(&(*temp)->right);
    }
    else{
        return temp;
    }
}


void popPengganti(data **temp){
    if((*temp)->left==NULL && (*temp)->right==NULL){
        free(*temp);
        *temp=NULL;
    }
    else if((*temp)->left!=NULL && (*temp)->right==NULL){
        data *temp2 = *temp;
        *temp = (*temp)->left;
        free(temp2);
    }
    else if((*temp)->right!=NULL && (*temp)->left==NULL){
        data *temp2 = *temp;
        *temp = (*temp)->right;
        free(temp2);
    }
    else if((*temp)->right!=NULL && (*temp)->left!=NULL){
```

```
74          data **curr = getAnakKiriPalingKanan(&(*temp)->left);
75          (*temp)->angka = (*curr)->angka;
76          popPengganti(curr);
77      }
78  }
79
80
81  void pop(data **curr,int angka){
82      if((*curr)->angka == angka){
83          popPengganti(curr);
84      }
85      else{
86          if(angka>(*curr)->angka)pop(&(*curr)->right,angka);
87          else if(angka<(*curr)->angka)pop(&(*curr)->left,angka);
88      }
89  }
90
91
92
93  int main(){
94      /*insertNode(&root,13);
95      insertNode(&root,15);
96      insertNode(&root,10);*/
97
98      /*insertNode(&root,30);
99      insertNode(&root,15);
100     insertNode(&root,7);
101     insertNode(&root,26);
102     insertNode(&root,19);
103     insertNode(&root,17);
104     insertNode(&root,21);
105     insertNode(&root,37);
106     insertNode(&root,34);
107     insertNode(&root,31);
108     insertNode(&root,45);
109     insertNode(&root,42);*/
110
111     insertNode(&root,15);
112     insertNode(&root,7);
113     insertNode(&root,9);
114     insertNode(&root,8);
115
116
117
118     pop(&root,9);
119     insertNode(&root,9);
120     //pop(&root,30);
121     printf("Preorder : ");
122     preorder(&root);
123     printf("\n");
124     printf("Inorder : ");
125     inorder(&root);
126     printf("\n");
127     printf("Postorder : ");
128     postorder(&root);
129     printf("\n");
130     getchar();
131     return 0;
132 }
133
```