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

//Membuat struct
struct tree{
    int data;
    char name[50];
    struct tree *left;
    struct tree *right;
};

//Inisialisasi variabel
struct tree *root = NULL;
int data;
int name;

//Fungsi utama
int main()
{
    display();
    return 0;
}

//Fungsi untuk tampilan program
void display(menu){
    int pilihan;
    printf("PINK LIBRARY\n");
    printf("************\n\n\n");
    printf("1. View All Book\n");
    printf("2. Add Book\n");
printf("3. Remove Book\n");
printf("4. Inorder, Preorder, Postorder\n");
printf("5. Exit and Remove All\n\n");
printf(">> Input choice : ");
scanf("%d",&menu);
if(menu == 1){
    view();
}
else if(menu == 2){
    add();
    root = createNode(root);
    insertNode(root, name, data);
}
else if(menu == 3){
    removeBook();
    deleteNode(root, name, data);
}
else if(menu == 4){
    printf(">> Input choice 1-3 : ");
    scanf("%d",&pilihan);
    if(pilihan == 1){
        inorder(root);
    }
    else if(pilihan == 2){
        preorder(root);
    }
    else if(pilihan == 3){
        postorder(root);
    }
    else{
        printf("Your choice not found\n\n");
        }
        return display();
    }
    else if(menu == 5){
        exit(0);
    }
    else{
        printf("Your choice not found\n\n");
        return display();
    }
}

//Fungsi untuk melihat data
void view(){
    if(root == NULL){
        printf("--- There is No Book in There ---\n\n");
    }
    else{
        printf("- %s (%d)", name, data);
    }
    return display();
}

//Fungsi untuk menambah data
void add(){
    getchar();
    printf("Input Book's Name [3..50]: ");
    scanf("%[^\n]",name);
    getchar();
    if(strlen(name) < 3 || strlen(name) > 50){
        printf("You Inputted Name Wrongly\n\n");
        return add();
    }
    else{
        printf("Input Book' s Number [0..100]: ");
        scanf("%d",&data);
        if(data < 0 || data > 100){
            printf("You Inputted Number Wrongly\n\n");
            return add();
        }
        else{
            printf("--- Add Book Success ---\n\n");
            return display();
        }
    }
}

//Fungsi untuk menghapus data
void removeBook(struct tree *root, int data){
    if(root == NULL){
        return root;
    }
    printf("Input Book' s Number [0..100]: ");
    scanf("%d",&data);
    if(data < 0 || data > 100){
            printf("You Inputted Number Wrongly\n\n");
        }
        else{
            printf("--- Delete Book Success ---\n\n");
            return display();
        }
}
```

```
//Struct untuk membuat node
struct tree *createNode(const char *name, int data) {
  struct tree *newNode = (struct tree*)malloc(sizeof(struct tree));
  strcpy(newNode->name,name);
  newNode->data= data;
  newNode->left = newNode->right = NULL;
  return newNode;
}

//Struct untuk menambah data
struct tree *insertNode(struct tree *root, const char *name ,int data){
  if(root == NULL){
    return createNode(data, name);
  }
  else if(data < root->data){
    root->left = insertNode(root->left, data, name);
  }
  else if(data > root->data){
    root->right = insertNode(root->right, data, name);
  }
  return root;
}

//Struct untuk menghapus data
struct tree *deleteNode(struct tree *root, int data){
    if (root == NULL){
        return root;
    }
    if (data < root->data){
        root->left = deleteNode(root->left, data);
    }
    else if (data > root->data){
        root->right = deleteNode(root->right, data);
    }
    else {
        if (root->left == NULL) {
            struct tree *newNode = root->right;
            free(root);
            return newNode;
        }
        else if (root->right == NULL) {
            struct tree *newNode = root->left;
            free(root);
            return newNode;
        }
    }
    return root;
}

//Fungsi untuk menampilkan data secara inorder, preorder, dan postorder
void inorder(struct tree *root){
    if(root != NULL){
        inorder(root->left);
        printf("%d\n",root->data);
        inorder(root->right);
    }
}

void preorder(struct tree *root){
    if(root != NULL){
        printf("%d\n",root->data);
        preorder(root->left);
        preorder(root->right);
    }
}

void postorder(struct tree *root){
    if(root!= NULL){
        postorder(root->left);
        postorder(root->right);
        printf("%d\n",root->data);
    }
}
```