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

struct BinaryTree
{
    struct BinaryTree *left;
    int data;
    struct BinaryTree *right;
};

struct BinaryTree * CreateBinaryNode(int value)
{
    struct BinaryTree *B=(struct BinaryTree *)malloc(sizeof(struct BinaryTree));
    B->left=NULL;
    B->right=NULL;
    B->data=value;
    return B;
};

void createMirrorImage(struct BinaryTree *root)
{
    if(root==NULL||(!root->left&&!root->right))
        return;
    else
    {
        createMirrorImage(root->left);
        createMirrorImage(root->right);
        struct BinaryTree *temp=root->left;
        root->left=root->right;
        root->right=temp;
    }

}

void preOrderTraversal(struct BinaryTree *root)
{
    if(root==NULL)
        return;
    printf("%d-->",root->data);
    preOrderTraversal(root->left);
    preOrderTraversal(root->right);
}

int main()
{
    struct BinaryTree *root;
    root=CreateBinaryNode(10);
    root->left=CreateBinaryNode(20);
    root->left->left=CreateBinaryNode(30);
    root->right=CreateBinaryNode(40);
    root->right->right=CreateBinaryNode(50);
    printf("Preorder Before Converting\n");
    preOrderTraversal(root);
    createMirrorImage(root);
    printf("\nPreorder After Converting\n");
    preOrderTraversal(root);
}
```