#include<stdio.h>

#include<stdlib.h>

struct node{

int key;

struct node\*left;

struct node\*right;

};

struct node\*newnode(int key)

{

struct node\*node=(struct node\*)malloc(sizeof(struct node));

node->left=NULL;

node->right=NULL;

node->key=key;

return node;

}

struct node\* insert(struct node\*root,int value)

{

if(root==NULL)

return newnode(value);

if(root->key>value)

root->left=insert(root->left,value);

if(root->key<value)

root->right=insert(root->right,value);

else

printf("root is already present");

return root;

}

struct node\* newtree(struct node \*root,int \*rightsum)

{

if(root==NULL)

return NULL;

root->right=newtree(root->right,&rightsum);

root->key=\*rightsum+root->key;

}

root->left=newtree(root->left,&rightsum);

return root;

}

void printpreorder(struct node\*root)

{

if(root==NULL)

return;

printf("%d",root->key);

printpreorder(root->left);

printpreorder(root->right);

}

void main()

{

struct node\*root;

root=NULL;

int a[1000],n;

printf("number of nodes in the tree:");

scanf("%d",&n);

printf("the binary tree:");

for(int i=0;i<n;i++)

{

scanf("%d",&a[i]);

root=insert(root,a[i]);

}

int rightsum;

rightsum=0;

root=newtree(root,&rightsum);

printpreorder(root);

}