struct BST{

int data;

struct BST \*left, \*right;

}node;

BST \*getNewNode()

{

BST \*tmp;

tmp=(BST \*) malloc(sizeof(BST));

tmp->left=NULL;

tmp->right=NULL;

return tmp;

}

void insertNode(BST \*root, BST \*newNode)

{

//cout<<root->data<<endl;

if(newNode->data < root->data)

{

if(root->left == NULL) root->left=newNode;

else insertNode(root->left, newNode);

}

else

{

if(root->right == NULL) root->right=newNode;

else insertNode(root->right, newNode);

}

return;

}

void preOrder(BST \*root)

{

cout<<root->data<<" ";

if(root->left !=NULL) preOrder(root->left);

if(root->right !=NULL) preOrder(root->right);

return;

}

void inOrder(BST \*root)

{

if(root->left !=NULL) inOrder(root->left);

cout<<root->data<<" ";

if(root->right !=NULL) inOrder(root->right);

return;

}

void postOrder(BST \*root)

{

if(root->left !=NULL) postOrder(root->left);

if(root->right !=NULL) postOrder(root->right);

cout<<root->data<<" ";

return;

}

BST \*searchBST(BST \*root, int key)

{

while(root!=NULL)

{

//cout<<root->data<<endl;

if(root->data == key) return root;

else if(key < root->data)

{

if(root->left != NULL)

root=root->left;

else return NULL;

}

else if(key > root->data)

{

if(root->right !=NULL)

root = root->right;

else return NULL;

}

}

return NULL;

}

int main()

{

BST \*root, \*newNode;

root = NULL;

for(int i=0;i<6;i++)

{

newNode = getNewNode();

cin>>newNode->data;

if(root==NULL) root=newNode; //BST is not initialized.

else insertNode(root, newNode);

}

return 0;

}