#include <stdio.h>

#include <stdlib.h>

struct n

{

int x;

struct n\* left;

struct n\* right;

};

struct n\* n1(int x)

{

struct n\*n = (struct n\*)

malloc(sizeof(struct n));

n->x = x;

n->left = NULL;

n->right = NULL;

return(n);

}

void printPostorder(struct n\*n)

{

if (n == NULL)

return;

printPostorder(n->left);

printPostorder(n->right);

printf("%d ", n->x);

}

void printInorder(struct n\*n)

{

if (n == NULL)

return;

printInorder(n->left);

printf("%d ", n->x);

printInorder(n->right);

}

void printPreorder(struct n\* n)

{

if (n == NULL)

return;

printf("%d ", n->x);

printPreorder(n->left);

printPreorder(n->right);

}

int main()

{

struct n\*root = n1(1);

root->left = n1(2);

root->right = n1(3);

root->left->left = n1(4);

root->left->right = n1(5);

printf("Preorder traversal on Binary Tree is :");

printPreorder(root);

printf("Inorder traversal on Binary Tree is :");

printInorder(root);

printf("Postorder traversal on Binary Tree is :");

printPostorder(root);

getchar();

return 0;

}