void inOrder(){

  // set current to root of binary tree

  Node \*current = root;

  Node \*s = NULL;

  bool done = 0;

  while (!done){

    // Reach the left most Node of the current Node

    if(current !=  NULL){

      // place pointer to a tree node on the stack before traversing

      //the node's left subtree

      push(&s, current);

      current = current->left;

    }

    // backtrack from the empty subtree and visit the Node at the top

// of the stack; however, if the stack is empty, you are done

    else{

      if (!isEmpty(s)){

        current = pop(&s);

        printf("%d ", current->data);

        // we have visited the node and its left subtree.

        // Now, it's right subtree's turn

        current = current->right;

      }else

        done = 1;

    }

  }

}

void postOrder () {

  if (!root) return;

  stack<Node\*> s;

  stack<Node\*> output;

  s.push(root);

  while (!s.empty()) {

    Node \*curr = s.top();

    output.push(curr);

    s.pop();

    if (curr->left) s.push(curr->left);

    if (curr->right) s.push(curr->right);

  }

  while (!output.empty()) {

    cout << output.top()->data << " ";

    output.pop();

  }

}