```cpp
#include <iostream>
#include <stack>

using namespace std;

struct Node {
    int data;
    Node *right;
    Node *left;

    Node(int data) : data(data) {
        right = left = nullptr;
    }
};

//traversing binary tree without recursion using stack.
void traverse(Node *root) {
    //making a stack
    stack<Node *> stc;
    //setting the current node as the root
    Node *current = root;
    while (current != nullptr || !stc.empty()) {
        //adding the current non-null node to the stack, then move to the left node of it.
        //do it more and more until we reach to the left most node
        while (current != nullptr) {
            stc.push(current);
            current = current->left;
        }
        //we found the leftmost node and saved it in stack, now we pop it from top of stack
and change the current node
        // to the top node of stack
        current = stc.top();
        stc.pop();
        //the top node of the stack is the one we should print here
        cout << current->data << endl;
        //after we traverse the left child of the current node we traverse the right part.
        //if the right part is null ,the above while will not run for it in the next run of the
outside while
        //it means that data of current node and its children (if exist) are printed and we move
up
        current = current->right;
    }
}
```

```cpp
int main() {

    //            ROOT(1)
    //           /       \
    //         (0)        (4)
    //         /         /   \
    //       (-1)       (2)   (5)
    //                   \
    //                   (3)

    //making a BST
    Node *root = new Node(1);
    root->left = new Node(0);
    root->left->left = new Node(-1);
    root->right = new Node(4);
    root->right->left = new Node(2);
    root->right->left->right = new Node(3);
    root->right->right = new Node(5);

    //testing out function
    traverse(root);
    return 0;
}
```