

pre-order tree traversal:-

(A)

int main()

{

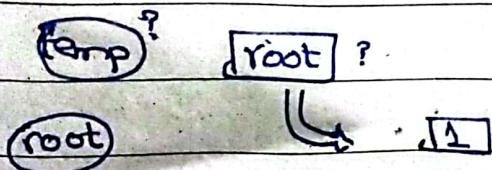
Struct Node *root = new node(1);

[root]

(The newnode function is called)

Node * newnode (int data)

{ Node * temp = new node;

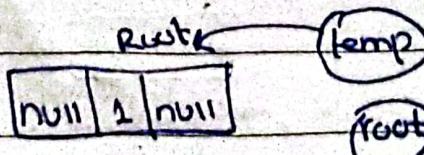


- temp → data = data;

temp → [1] root node

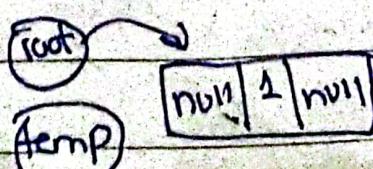
- temp → left = temp → right = Null;

- return temp ;

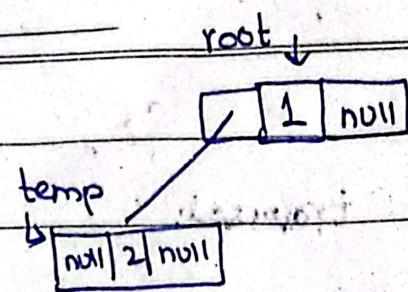


(again going to main function)

- root → left = new node(2);



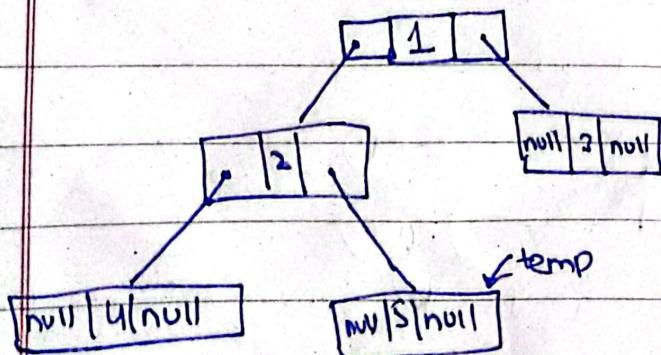
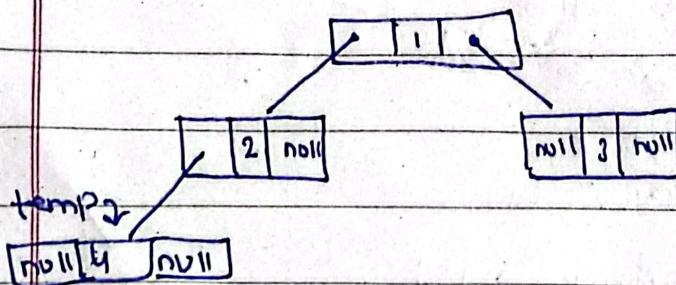
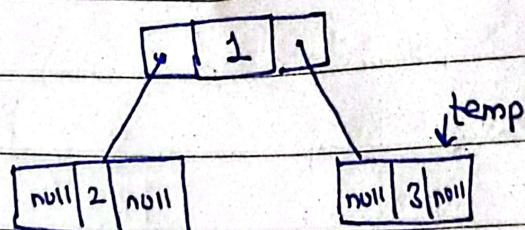
Date: _____



- $\text{root} \rightarrow \text{right} = \text{new node}(3);$

(same procedure repeated)

(now temp will point to right)

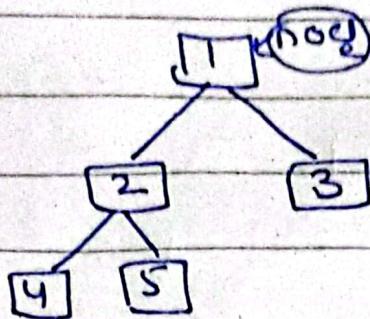


print preorder (root);

(now printpreorder function
is called)

if condition doesn't apply so..)

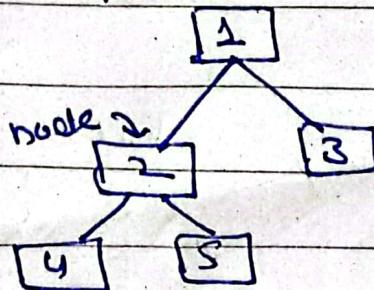
- cout << node->data << " ";



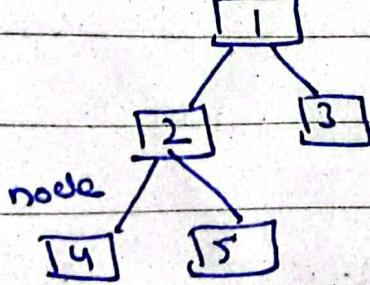
This pointer From
void printpreorder (struct node*)

By this, node's data is printed(1)
printpreorder (node->left);

root 2



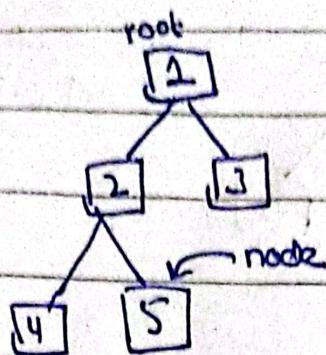
again the whole
function is called
for left node.



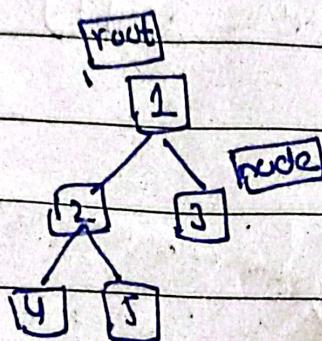
output: 1, 2, 4

Now, this node "4" doesn't
has left subtree (if condition
applied) so, Function returns

- print preorder (node → right);



output : 1 2 4 5



function is
called again

output : 1 2 4 5 3

After this, the main function
is terminated by return 0.

Post-order traversal:

- int main () {

 struct node* root = new node(1);

 root → left = new node(2);

 root → right = new node(3);

 root → left → left = new node(4)

Date: _____

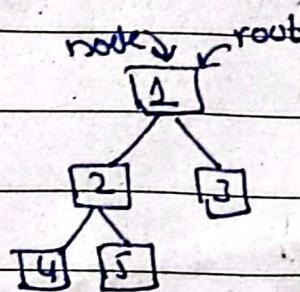
root \rightarrow left \rightarrow right = new node ('5');

(Tree is created by same procedure as previous code)

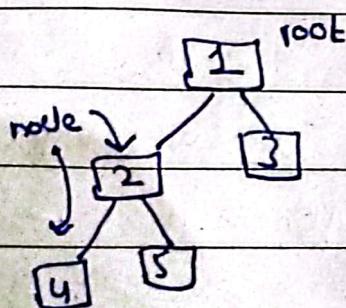
- printpostorder (root);

if - condition is not applicable
so function proceeds to next

line



- printpostorder (node->left);

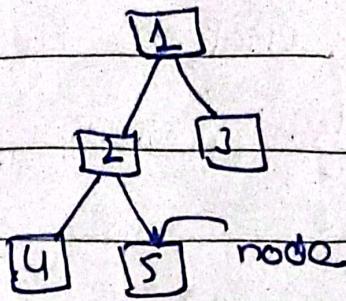


Now "4" node doesn't have
left and right subtree,
so its data is printed

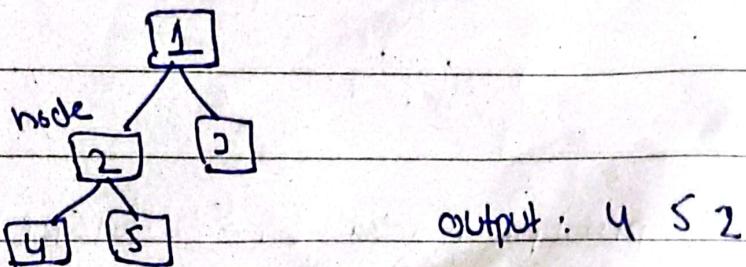
'in output by

cout << node->data << " "

- printpostorder (node → right);



now, this "5" doesn't have
any left or right subtree,
so, its data will be printed
in output.

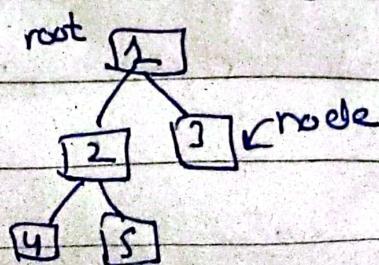


Again Function calls:

- printpostorder (node → right);

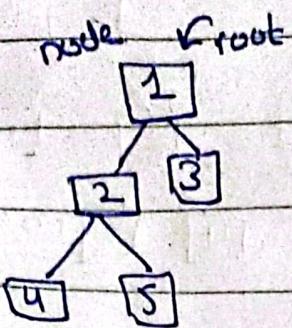
now, in function, node
pointers

points to right of root



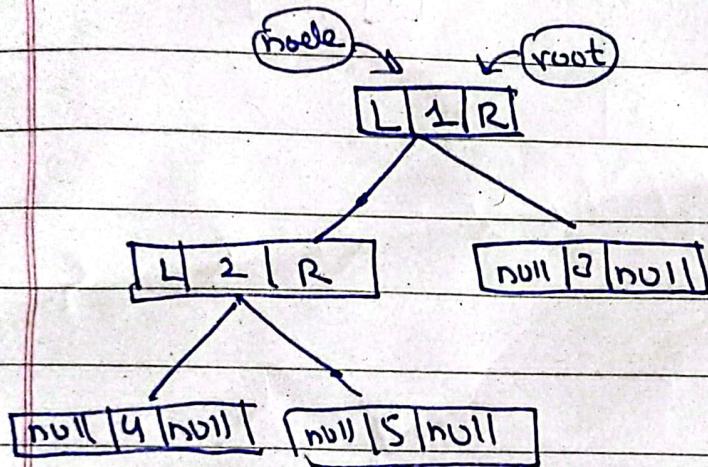
Date: _____

output : 4 5 2 3

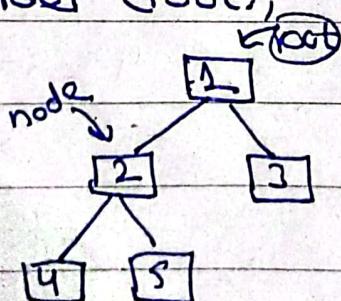


output : 4 5 2 3 1

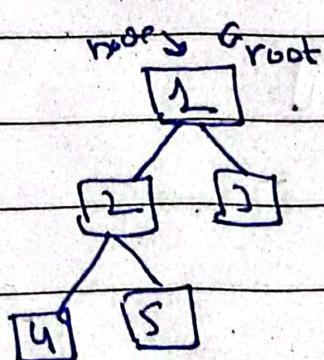
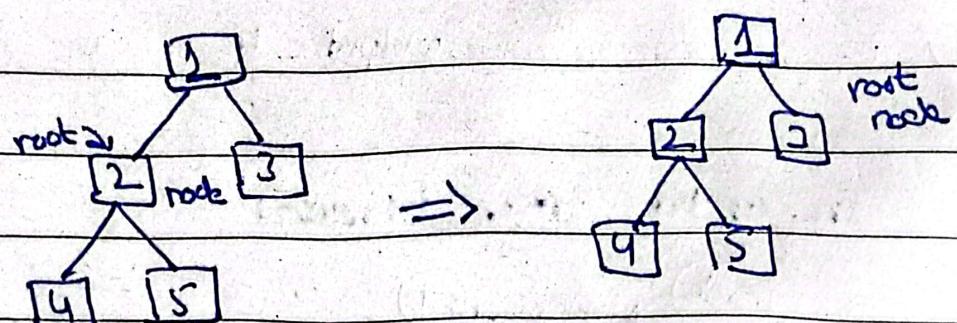
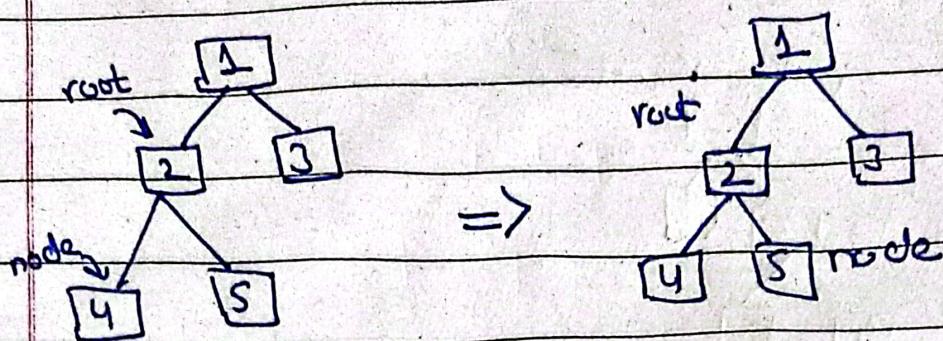
In-order tree traversal



- print_order (root);



Date: _____

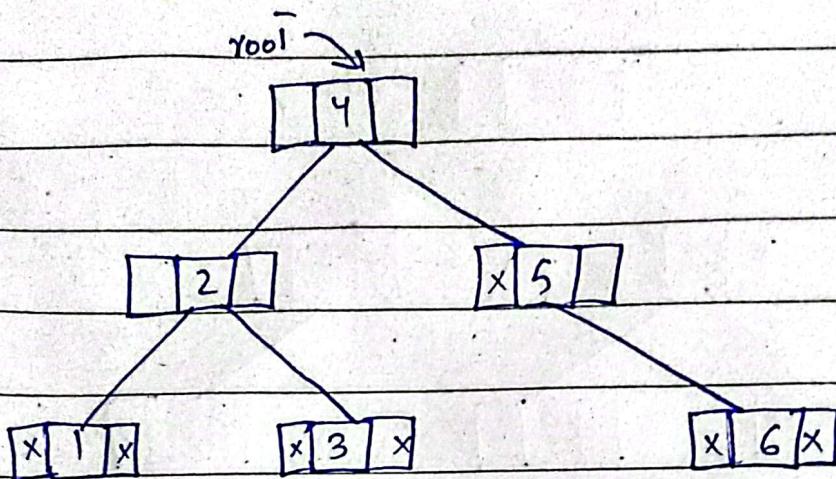


Output:

4 2 5 1 3

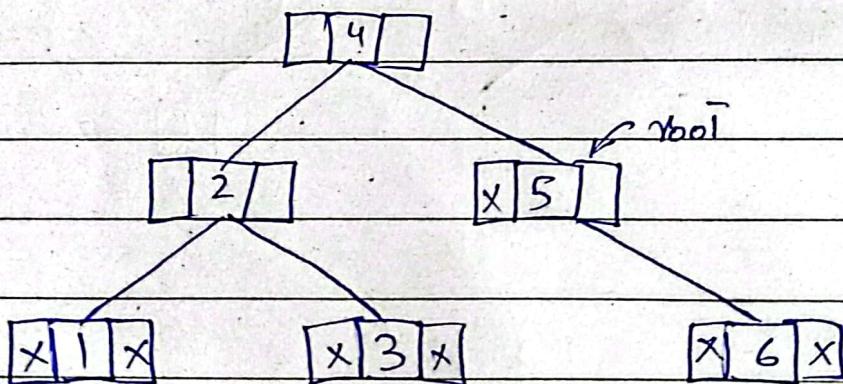
Date: _____

Searching in BST:



Value to be searched = key = 5.

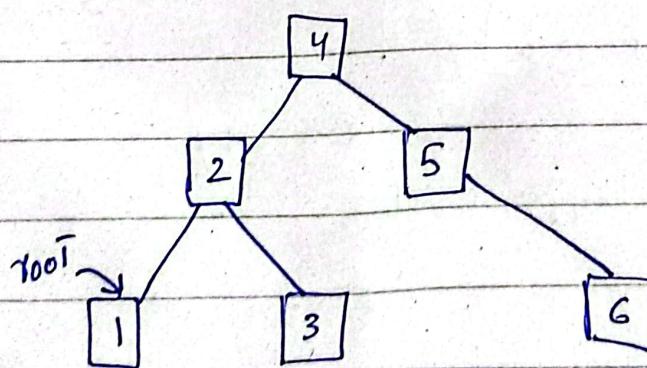
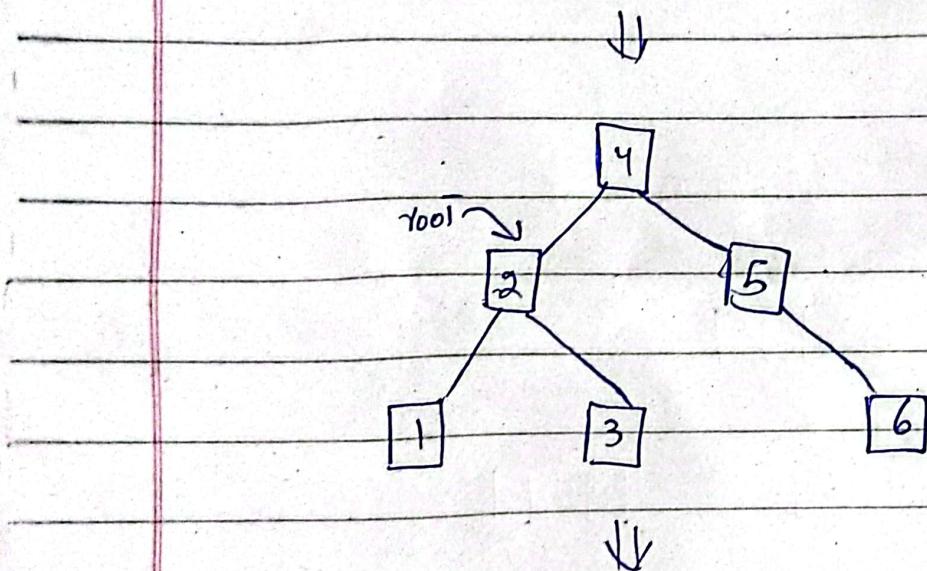
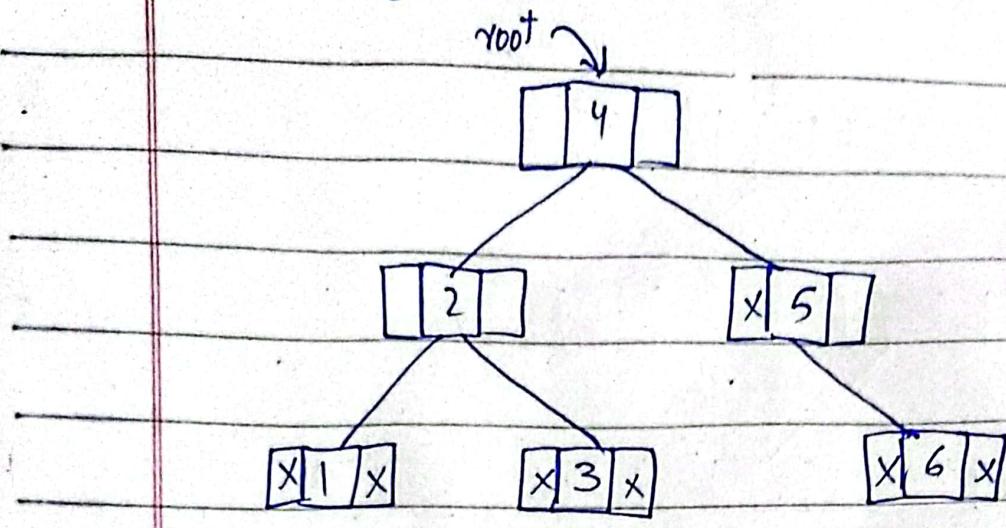
$\text{root} \rightarrow \text{data} = \text{key}$



$\text{root} \rightarrow \text{data} == \text{key}$

"key is found"

Searching minimum value in BST:



"min-value
found."