LAB Assigment

Name : Muhammad Abdullah

Registration: Sp22 - BCS - 036

Subject: DSA

Submmited To: yasmeen Jana.

CUI vehari Campus

→ Dequeue : Return the top of S_1 which is 2 and pop it

S_1 : 3

S_2 :

→ Dequeue : Return the Top S_1 which is 3 and pop it

S_1 :

S_2 :

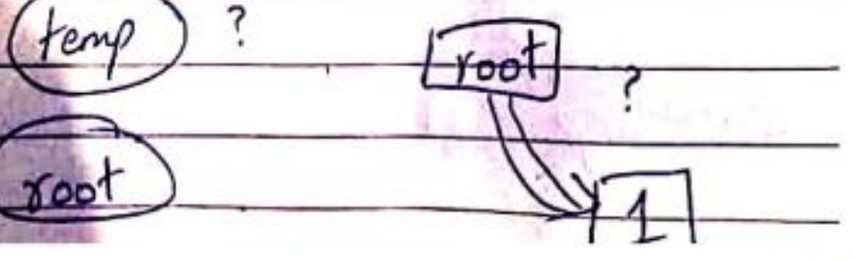
- **Pre-order : tree traversal :**

```
- int main( )
- {
- Struct Node root = new Node (1)
```

root

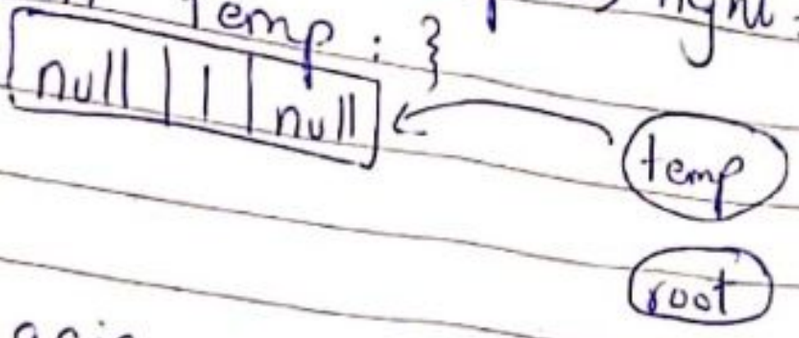
(The new node function is called)

```
- Node * new node (int data)
- { Node * temp = new Node;
```



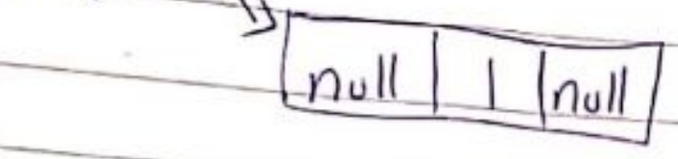
CamScanner

- temp → data = data;
  temp → 1 (root node)
- temp → left = temp → right = Null;
- return temp; }



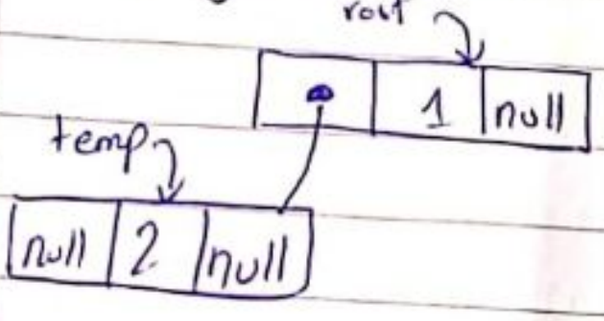
(again going to main function)

root → left = new node(2);

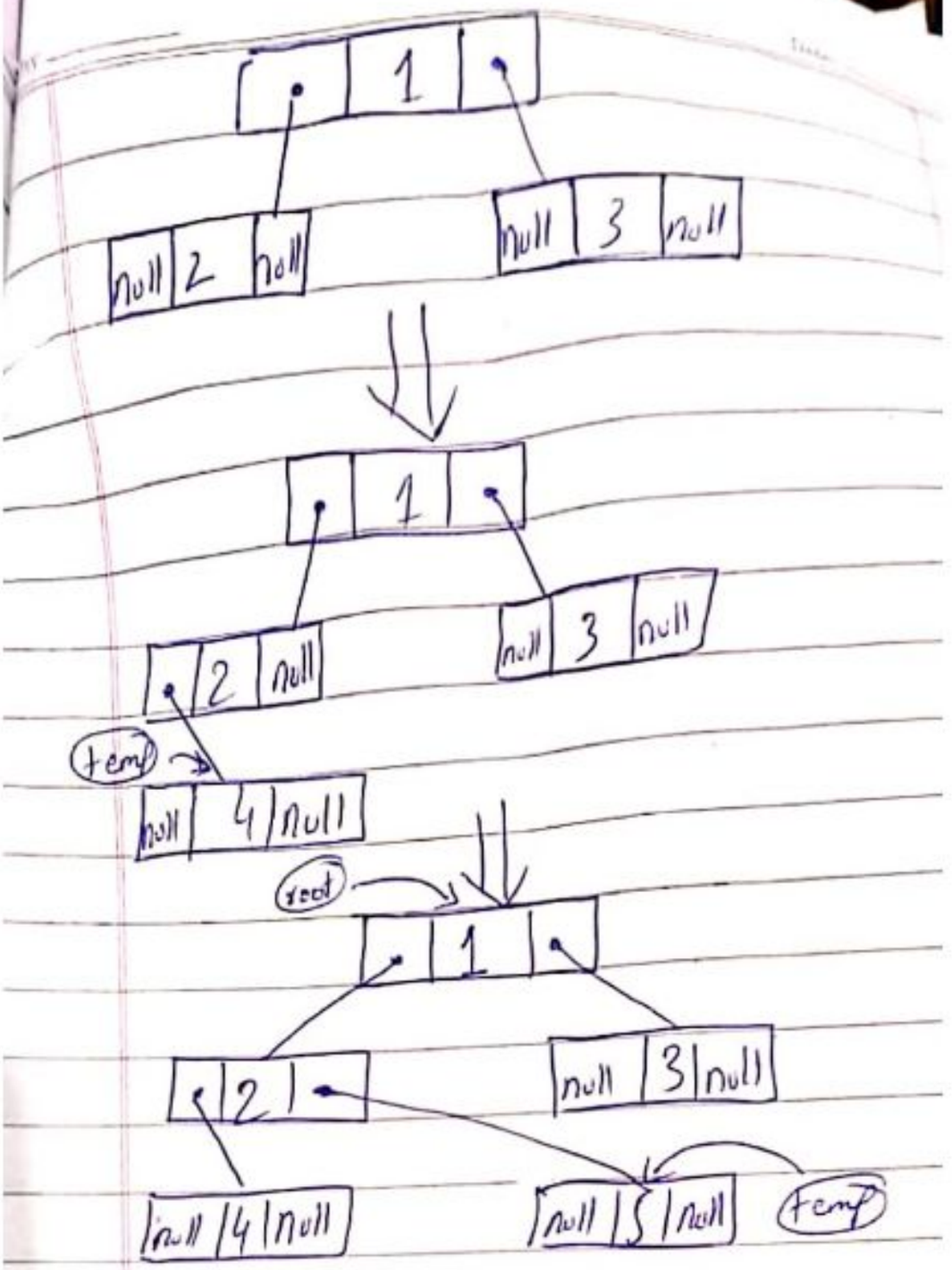


temp

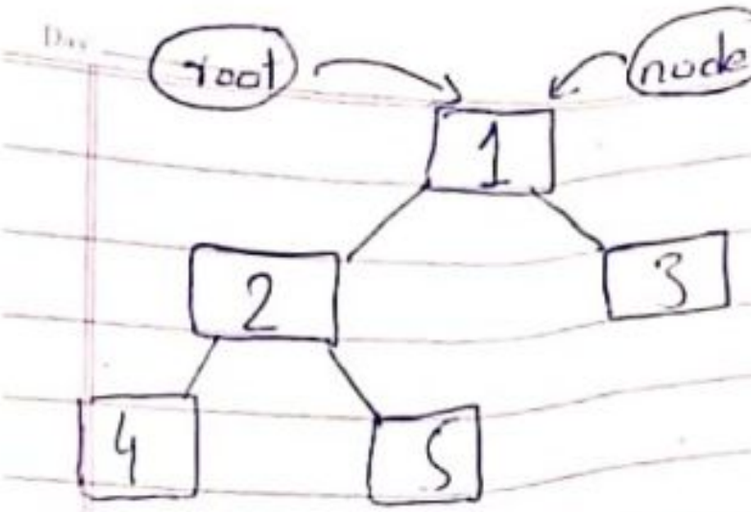
(again new node function called)



- root → right = new Node(3);
  (Same procedure repeated, now temp with point to right)



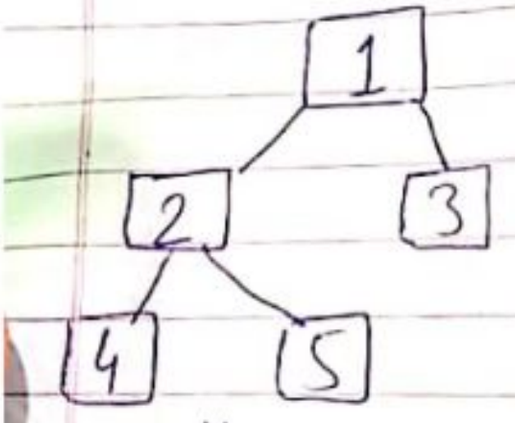
- Print pre-order (root):
  (new print preorder function is called)
  (if condition does not apply so...)
- cout << node -> data << " ";



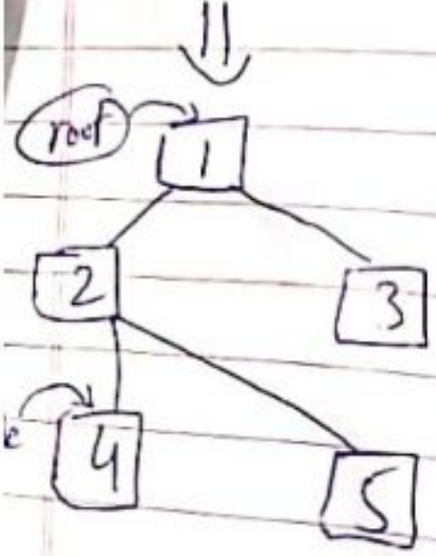
→ Thus pointer from [void print pre order (struct node node)

By this node's data is printed (1)
output : 1

- 1 print pre order (node → left);



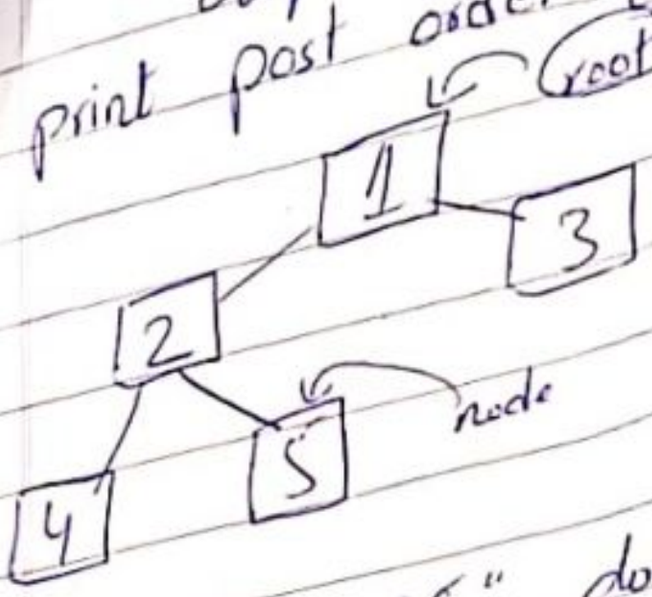
(again the whole function is called for left node)
Output : 1 2



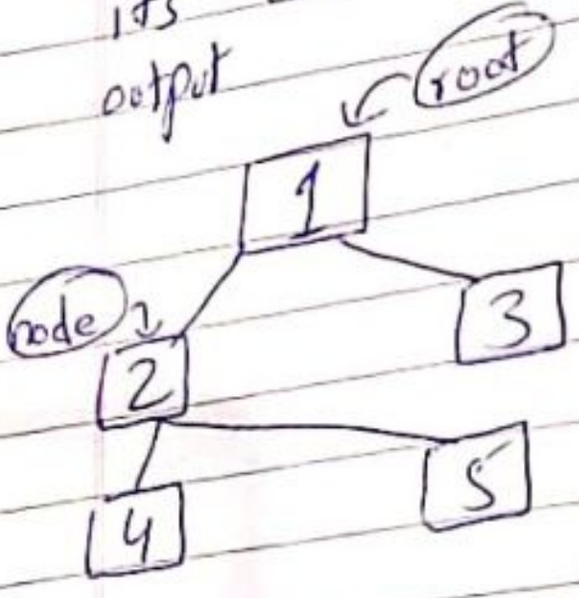
Output : 1 2 4

Output : 4

- print post order (node → right);



Now, this "5" does not has any left or right subtree, so; its data will be printed in output



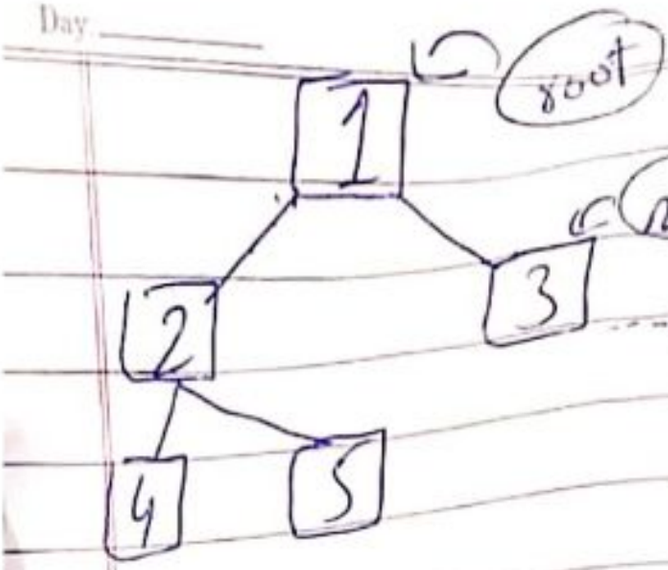
Output : 4 5

Output : 4 5 2

Again, function calls:

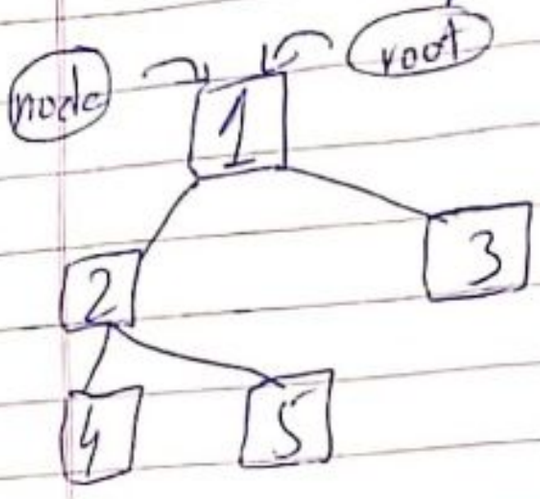
Print post order (node → right);

Now, in function, node pointer ... to right of root.



(does not has left or right subtree)

Output : 4 5 2 3

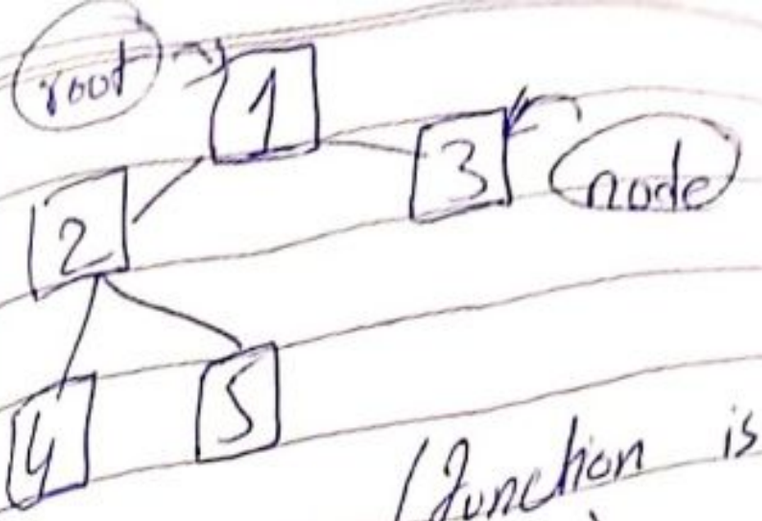


Output : 4 5 2 3 1

Now, this node "4" does not has left subtree (if condition applied so function returns

- print preorder (node -> right);

Output : 1 2 4 5



(function is called again)

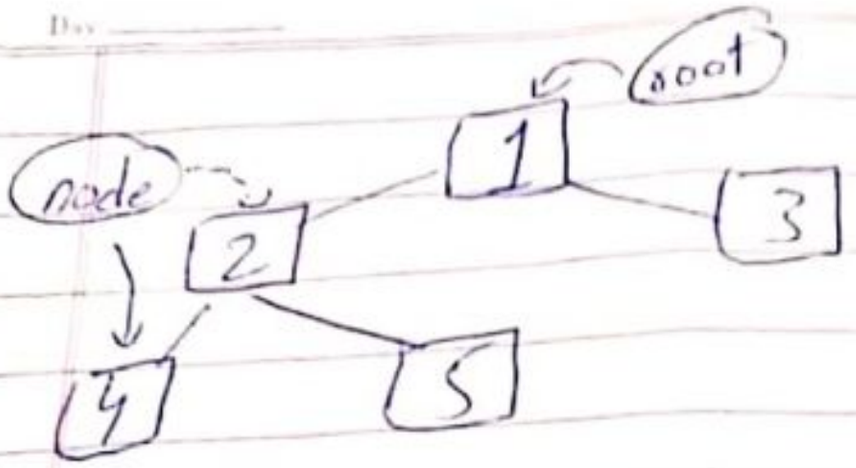
Output : 1 2 4 5 3

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

• Post-order tree 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);

  (Tree is created by same procedure
  as previous code)
- Print Post order (root);
     (Print Post order function is called)
  if condition is not applicable
```



Now "4" node does not has left & right subtree so, its data is printed in output by
- cout << node -> data << " ";

- Print post order function proceed to next line