	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESMENT YEAR: 2020-2021	CLASS: SE 5
	SUBJECT: DATA STRUCTURES	
EXPT No: 5	LAB Ref: SE/2020-21/	Starting date: 26/11/2020
	Roll No: 22119	Submission date:03/12/2020
Title:	Creation of BST	

Prerequisites:	Programmer must be aware of concepts of dynamic memory allocation
	Programmer must be aware of forming tree using struct
	Programmer must be aware of basic C syntaxes
	Programmer must be able to write, build and run code in DEVC++
Objectives:	• To learn the concepts of nonlinear Data Structure (Non cyclic data structure), Apply it in the creation of Binary Search Tree (BST).
	• Implement different algorithms traversing to BST.

Theory:

A Binary Search Tree (BST) is a tree in which all the nodes follow the below-mentioned properties −

- The value of the key of the left sub-tree is less than the value of its parent (root) node's key.
- The value of the key of the right sub-tree is greater than or equal to the value of its parent (root) node's key.

Thus, BST divides all its sub-trees into two segments; the left sub-tree and the right sub-tree and can be defined as −

left_subtree (keys) < node (key) ≤ right_subtree (keys)

Basic Operations

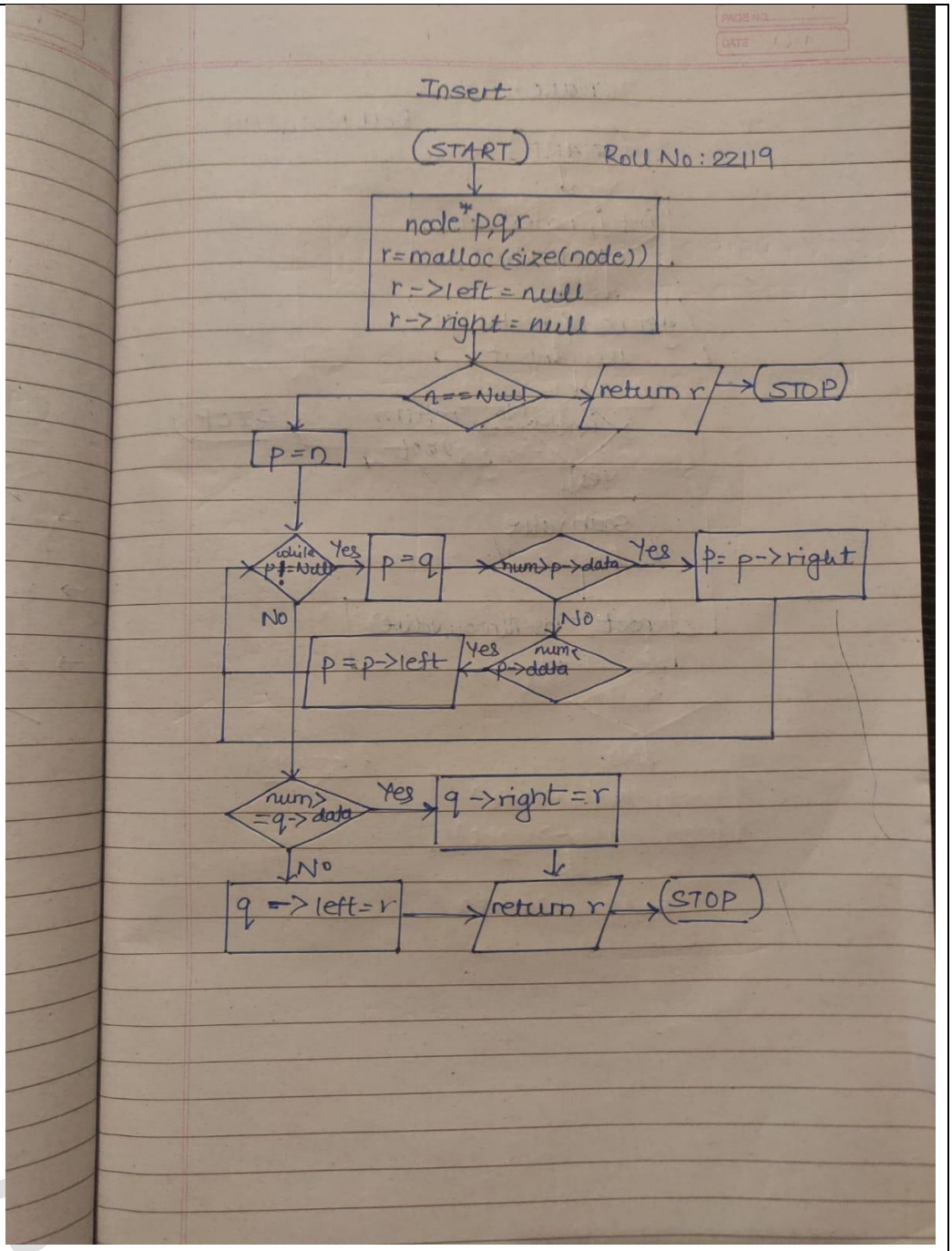
Following are the basic operations of a tree –

- **Search** − Searches an element in a tree.
- **Insert** − Inserts an element in a tree.
- **Pre-order Traversal** − Traverses a tree in a pre-order manner.
- **In-order Traversal** − Traverses a tree in an in-order manner.
- **Post-order Traversal** − Traverses a tree in a post-order manner.

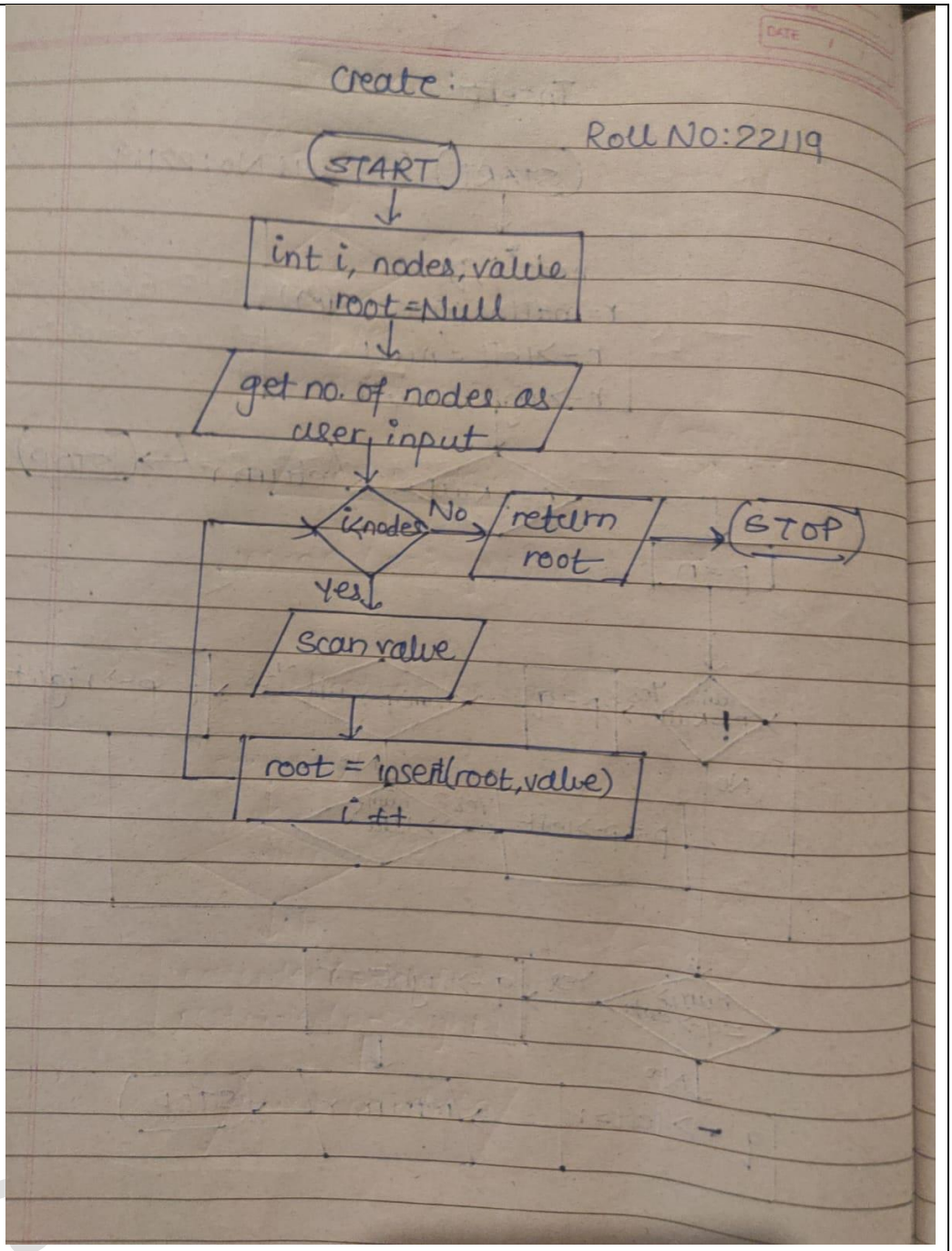
Algorithm	Insert Algorithm- 1. Start. 2. Declare 3 pointer to node p,q,r. 3. Allocate memory for pointer r, make left and right null and add the data provided by user into data field. 4. If Given node n is equal to NULL then return r and go to step 13. 5. P = n (given node). 6. If p== NULL go to step 10 else step 7. 7. q= p. 8. If number > p -> data then make p = p -> right. And go to step 6. 9. If number < p -> data then make p = p -> left. And go to step 6. 10. If number >= q -> data, make q -> right = r and go to step 12. 11. If number < q -> data, make q -> left = r and go to step 12. 12. Return n (given node). 13. Stop. Create Algorithm- 1. Start. 2. Declare int i,nodes,value. Initialize node* root = NULL. 3. Scan number of nodes from user and store it in nodes. 4. If i > nodes go to step 8 else step 5. 5. Scan value from user and store it into value. 6. Call insert function as, root = insert(root,value). 7. Go to step 4. 8. Stop. Find Algorithm- 1. Start. 2. If root == NULL go to step 6. 3. If root -> data = number, then return root and go to step 7. 4. If number > root -> data, then root = root -> right and go to step 2 5. If number < root -> data then root = root -> left and go to step 2. 6. Return NULL. 7. Stop. Preorder Algorithm- 1. Start. 2. If p == NULL then go to step 6. 3. Print p -> data. 4. Preorder(p -> left). 5. Preorder(p- > right). 6. Stop. Inorder Algorithm - 1. Start. 2. If p == NULL then go to step 6. 3. Inorder(p -> left). 4. Print p -> data.

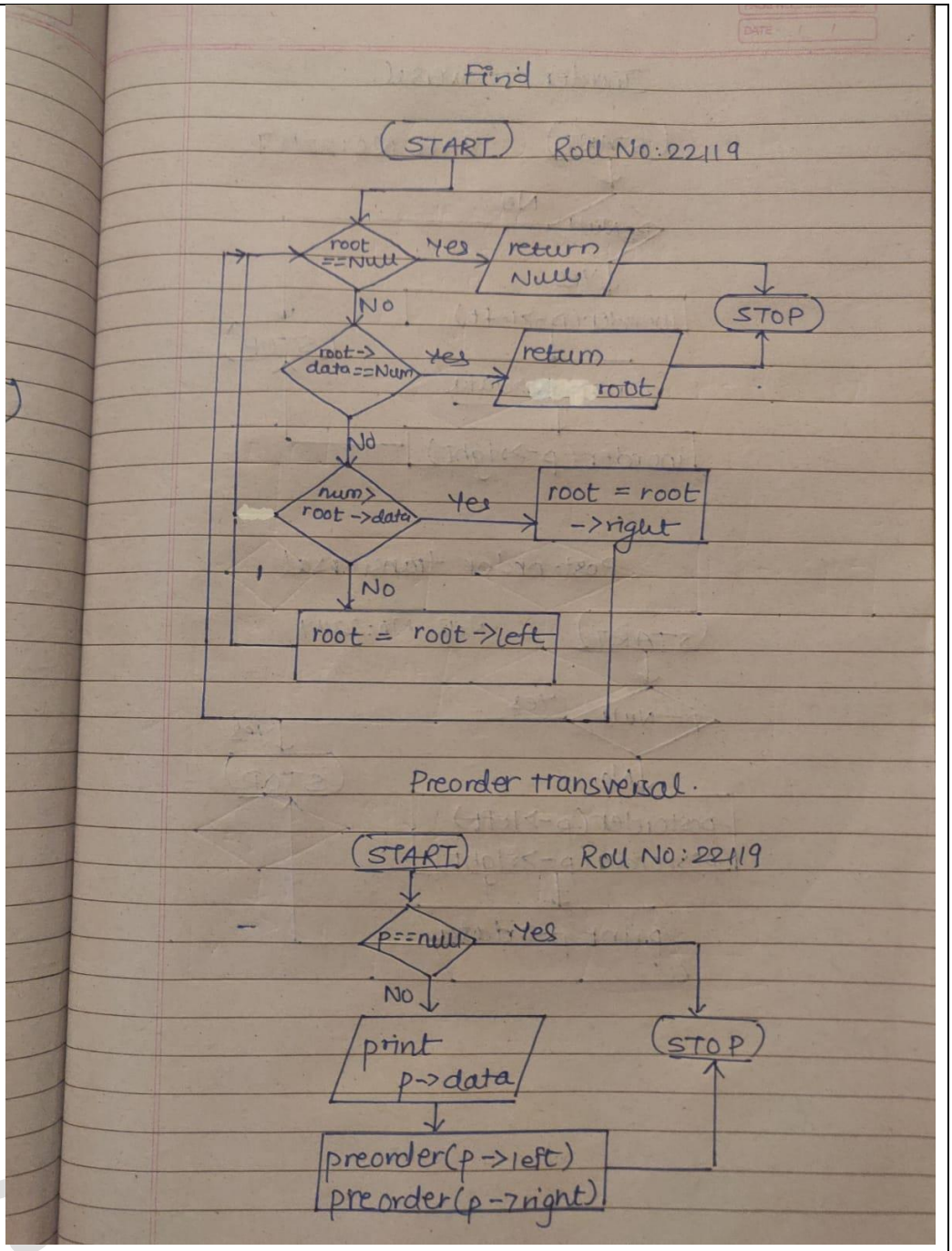
5. Inorder(p -> right).
6. Stop.

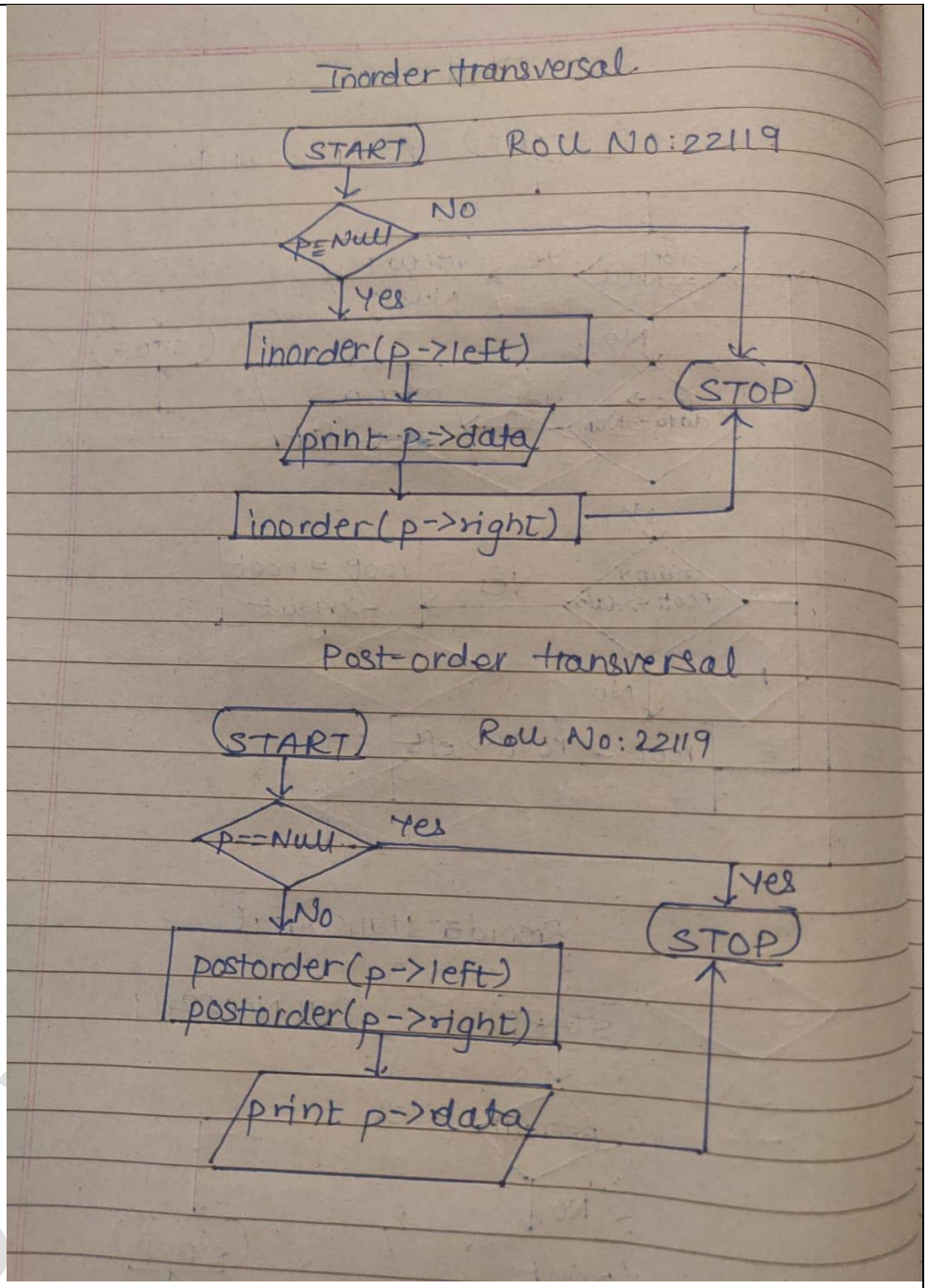
Postorder Algorithm-
1. Start.
2. If p == NULL then go to step 6.
3. Postorder(p -> left).
4. Postorder([p -> right).
5. Print(p -> data).
6. Stop.

Flow-chart	

Insert
START
Roll No: 22119
node* p,q,r
r=malloc(size(node))
r->left = null
r->right = null
r==Null
return r
STOP
p=r
while p!=Null
Yes
p=q
num>p->data
Yes
p= p->right
No
No
p=p->left
Yes
num<p->data
num>=q->data
Yes
q->right=r
No
q->left=r
return r
STOP


Create:
Roll No:22119
START
int i, nodes, value
root=Null
get no. of nodes as
user input
i<nodes
No
return
root
STOP
yes
scan value
root = insert(root, value)
i ++


Find
START
Roll No: 22119
root == Null
Yes
return Null
No
STOP
root -> data == Num
Yes
return root
No
num > root -> data
Yes
root = root -> right
No
root = root -> left
Preorder transversal.
START
Roll No: 22119
p == null
Yes
No
print p -> data
STOP
preorder(p -> left)
preorder(p -> right)


Inorder transversal
START
Roll No:22119
p==Null
No
Yes
inorder(p->left)
print p->data
inorder(p->right)
STOP
Post-order transversal
START
Roll No:22119
p==Null
Yes
No
Yes
postorder(p->left)
postorder(p->right)
print p->data
STOP

ERROR	None
REMEDY	None
CONCLUSION:	
	• learnt the concepts of nonlinear Data Structure • binary search tree created • Implementatiom of different algorithms traversing to BST id done
REFERENCES:	
	1. Ellis Horowitz, Sartaj Sahani, "Fundamentals of Data Structures", Galgotia books. 2. Richard F. Gilberg and Behrouz A. Forouzan, Data Structures A Pseudo code approach with C, cengage learning, 2nd edition. 3. Yashvant Kanetkar-Understanding Pointers in C BPB publications 3rd Edition.

Continuous Assessment			**Assessed By**
RPP (5)	**ARR (5)**	**Total (10)**	**Signature:**
			Date: