Last week program:- Program - 9

-> traversal

```
Struct node {
        int data;
        Struct node* left;
        Struct node* right;
}
```

-> traversal -> InorderTraversal

```
inorder (root->left)
display (root->data)
inorder (root->right)
```

Pseudo code

```
void inorderTraversal (struct node* root){
if(root==NULL) return;
inorderTraversal (root->left);
  Print("%d->", root->item);
  inorderTraversal (root->right);
```

-> Pre order Traversal

```
display (root->data)
preorder(root->left)
preorder (root->right)
```

Pseudo code.

```
void preorderTraversal (struct node* root){
    if (root == NULL) return;
    preorderTraversal (root->left);
    preorderTraversal(root->right);
    Printf ("%d->", root->item);
```

-> Postorder Traversal

```
Postorder (root->left)
Postorder (root->right)
display (root->data).
```

Pseudo Code -> ~~void~~

~~if~~:

```
void postorderTraversal (struct node* root) {
    if (root==NULL) return;
    postorderTraversal (root->left);
    postorderTraversal (root->right);
    printf ("%d->", root->item);
}
```

Creating Tree Node:

```
→ struct node* createNode (value) {

    Struct node* newNode = malloc( sizeof (struct node));
      newNode -> item = value;
      newNode -> left = NULL;
       newNode -> right = NULL;
        return newNode;
  }
```

Insertion on the right of the Node.

```
Struct node * insertRight (struct node* root, int value)
{
    root ->right = CreateNode(value);
     return root -> right;
}
```

Insertion on the left of the node.

```
Struct node * insertLeft( struct node* root, int value)
{
    root-> left = CreateNode(value);
      return root -> left;
}
```