3) Write a program

(a) To construct a binary search tree.

(b) To traverse the tree using all the methods i.e. in-order, preorder and post order.

(c) To display the element in the tree.

```
#include <stdio.h>
#include <stdlib.h>

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

struct Node *createNode(int data) {
    struct Node *newNode = (struct Node*) malloc(
                                sizeof(struct Node));
    newNode->data = data;
    newNode->left = NULL;
    newNode->right = NULL;
    return newNode;
}

struct Node* insert(struct Node *root, int data){
    if(root == NULL) {
        return createNode(data);
    }
    if(data < root->data) {
        root->left = insert(root->left, data);
    }
    else if(data > root->data) {
        root->right = insert(root->right, data);
```

```
    return root;
}
void inorderTraversal(struct Node* root) {
    if (root != NULL) {
        inorderTraversal(root->left);
        printf("%d", root->data);
        inorderTraversal(root->right);
    }
}
void postorderTraversal(struct Node* root) {
    if (root != NULL) {
        postorderTraversal(root->left);
        postorderTraversal(root->right);
        printf("%d", root->data);
    }
}
void preorderTraversal(struct Node* root) {
    if (root != NULL) {
        printf("%d", root->data);
        preorderTraversal(root->left);
        preorderTraversal(root->right);
    }
}
void displayTree(struct Node* root) {
    if (root != NULL) {
        printf("\n Inorder traversal: ");
        inorderTraversal(root);
        printf("\n Postorder traversal: ");
        postorderTraversal(root);
        printf("\n Preorder traversal: ");
```

```
        preorderTraversal(root);
    }
}

int main() {
    struct Node* root = NULL;
    root = insert(root, 30);
    displayTree(root);
    return 0;
}
```

Output :-

Inorder traversal : 30

Postorder traversal : 30

Preorder traversal : 30

Hacker Rank Challenge

Merging of two sorted lists :-

Input

```
1
3
1
2
3

2
3
4
```

Your Output → 1 2 3 3 4

Expected Output → 1 2 3 3 4

19.02.24