Practical Number -17

Aim: To write a program in C and Java to perform **Preorder Traversal of a Binary Tree**.

Theory

Tree traversal is the process of visiting each node of a tree exactly once in a specific order. In a **binary tree**, traversal can be performed using different methods such as preorder, inorder, and postorder traversal.

In Preorder Traversal, nodes are visited in the following order:

Root → Left Subtree → Right Subtree

Preorder traversal is a Depth-First Search (DFS) technique and is commonly implemented using recursion.

Steps of preorder traversal:

1. Visit the root node.
2. Traverse the left subtree.
3. Traverse the right subtree.

C Code

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

// Structure of a node
struct node {
    int data;
    struct node* left;
    struct node* right;
};

// Function to create tree using user input
struct node* createTree() {
    int data;

    printf("Enter data (-1 for no node): ");
    scanf("%d", &data);

    // Base condition
    if (data == -1)
        return NULL;

    // Create new node
    struct node* newNode = (struct node*)malloc(sizeof(struct node));
    newNode->data = data;

    printf("Enter left child of %d\n", data);
    newNode->left = createTree();

    printf("Enter right child of %d\n", data);
    newNode->right = createTree();

    return newNode;
}

// Preorder Traversal
void preorderTraversal(struct node* root) {
    if (root == NULL)
        return;

    printf("%d ", root->data);
    preorderTraversal(root->left);
    preorderTraversal(root->right);
}

// Main function
int main() {
    printf("Create Binary Tree\n");
    struct node* root = createTree();

    printf("\nPreorder Traversal is: ");
    preorderTraversal(root);

    return 0;
}
```