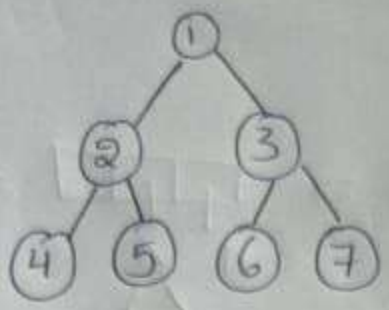
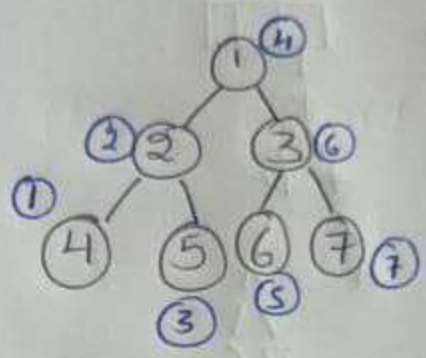
Problem for Binary Tree-Transversal:

BT :->

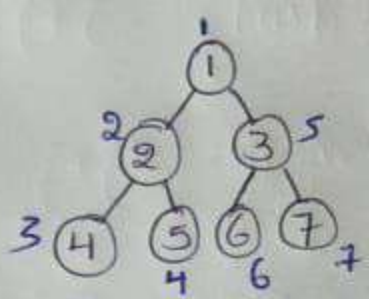


In order:



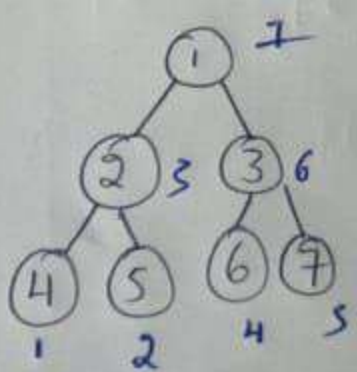
4 2 5 1 6 3 7

Pre Order:



1 2 4 5 3 6 7

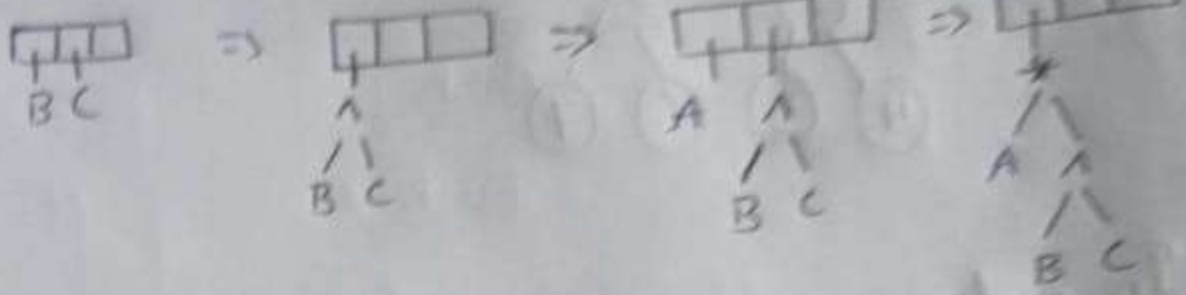
Post Order:

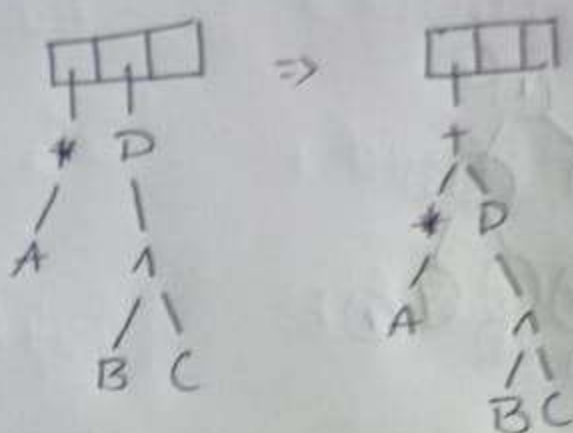


4 5 2 6 7 3 1

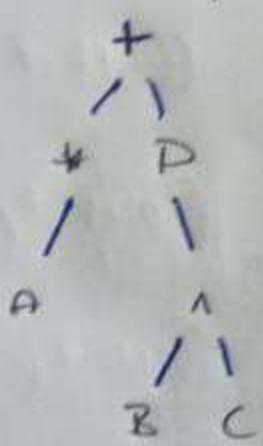
Binary Tree Expression

1) $A^{+}B\hat{}C+D$





BT - Expression



ChatGPT | Online C Compiler - Programiz | FREE AI Code Generator: Gener

programiz.com/c-programming/online-compiler/

Paused

Programiz
C Online Compiler

Programiz PRO ›

main.c

Share | Run

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

struct Node {
    int key;
    struct Node *left, *right;
};

struct Node* insert(struct Node* root, int key) {
    if (root == NULL) {
        struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
        newNode->key = key;
        newNode->left = newNode->right = NULL;
        return newNode;
    }

    if (key < root->key)
        root->left = insert(root->left, key);
    else if (key > root->key)
        root->right = insert(root->right, key);

    return root;
}

struct Node* search(struct Node* root, int key) {
    if (root == NULL || root->key == key)
        return root;

    if (root->key < key)
        return search(root->right, key);

    return search(root->left, key);
}

struct Node* minValueNode(struct Node* node) {
    struct Node* current = node;
```

Output | Clear

```
/tmp/bioTTbwkKV.o
Inorder traversal of the BST: 20 30 40 50 60 70 80
Delete 20
Inorder traversal of the modified BST: 30 40 50 60 70 80

=== Code Execution Successful ===
```

USD/EUR +0.34%

Search

ENG IN

8:53 PM 7/29/2024

Programiz
C Online Compiler

Programiz PRO ›

main.c

Share | Run

```
    while (current && current->left != NULL)
        current = current->left;

    return current;
}

struct Node* deleteNode(struct Node* root, int key) {
    if (root == NULL)
        return root;

    if (key < root->key)
        root->left = deleteNode(root->left, key);
    else if (key > root->key)
        root->right = deleteNode(root->right, key);
    else {
        if (root->left == NULL) {
            struct Node* temp = root->right;
            free(root);
            return temp;
        } else if (root->right == NULL) {
            struct Node* temp = root->left;
            free(root);
            return temp;
        }

        struct Node* temp = minValueNode(root->right);
        root->key = temp->key;
        root->right = deleteNode(root->right, temp->key);
    }

    return root;
}

void inorder(struct Node* root) {
    if (root != NULL) {
        inorder(root->left);
        printf("%d ", root->key);
```

Output | Clear

```
/tmp/bioTTbwkKV.o
Inorder traversal of the BST: 20 30 40 50 60 70 80
Delete 20
Inorder traversal of the modified BST: 30 40 50 60 70 80

=== Code Execution Successful ===
```

main.c

Share | Run

```
63          struct Node* temp = minValueNode(root->right);
64          root->key = temp->key;
65          root->right = deleteNode(root->right, temp->key);
66      }
67
68      return root;
69  }
70
71  void inorder(struct Node* root) {
72      if (root != NULL) {
73          inorder(root->left);
74          printf("%d ", root->key);
75          inorder(root->right);
76      }
77  }
78
79  int main() {
80      struct Node* root = NULL;
81      root = insert(root, 50);
82      insert(root, 30);
83      insert(root, 20);
84      insert(root, 40);
85      insert(root, 70);
86      insert(root, 60);
87      insert(root, 80);
88
89      printf("Inorder traversal of the BST: ");
90      inorder(root);
91
92      printf("\nDelete 20\n");
93      root = deleteNode(root, 20);
94      printf("Inorder traversal of the modified BST: ");
95      inorder(root);
96
97      return 0;
98  }
99
```

Output | Clear

```
/tmp/bioTTbwkKV.o
Inorder traversal of the BST: 20 30 40 50 60 70 80
Delete 20
Inorder traversal of the modified BST: 30 40 50 60 70 80

=== Code Execution Successful ===
```