

### Задание. Вариант 7.

Напечатать дерево в обычном виде и в виде: узел (левое поддерево) (правое поддерево)

### Реализация на языке C.

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

// Структура для узла дерева
typedef struct Node {
    int data;
    struct Node *left;
    struct Node *right;
} Node;

// Функция для создания нового узла дерева
Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->left = NULL;
    newNode->right = NULL;
    return newNode;
}

// Вставка узла в дерево
Node* insertNode(Node* root, int data) {
    if (root == NULL) {
        return createNode(data);
    }
    if (data < root->data) {
        root->left = insertNode(root->left, data);
    } else {
        root->right = insertNode(root->right, data);
    }
    return root;
}

// Печать дерева в обычном виде (inorder traversal)
void printInOrder(Node* root) {
    if (root != NULL) {
        printInOrder(root->left);
        printf("%d ", root->data);
        printInOrder(root->right);
    }
}

// Печать дерева в формате "узел (левое поддерево) (правое поддерево)"
void printTree(Node* root) {
    if (root != NULL) {
        printf("%d", root->data);
        if (root->left != NULL || root->right != NULL) {
            printf(" (");
            if (root->left != NULL) {
                printTree(root->left);
            } else {
                printf("NULL");
            }
        }
    }
}
```

```

    }
    printf(")");
    if (root->right != NULL) {
        printTree(root->right);
    } else {
        printf("NULL");
    }
    printf("");
}
}
}

int main() {
    Node* root = NULL;
    // Пример вставки в дерево
    int arr[] = {5, 3, 8, 1, 4, 7, 10};
    int n = sizeof(arr)/sizeof(arr[0]);
    for (int i = 0; i < n; i++) {
        root = insertNode(root, arr[i]);
    }

    printf("Дерево в обычном виде (inorder traversal):\n");
    printInOrder(root);
    printf("\n");

    printf("Дерево в формате \"узел (левое поддерево) (правое поддерево)\":\n");
    printTree(root);
    printf("\n");

    return 0;
}

```

## Пример работы программы.

Взаимодействие с программой:

```

Дерево в обычном виде (inorder traversal):
1 3 4 5 7 8 10
Дерево в формате "узел (левое поддерево) (правое поддерево)":
5 (3 (1) (4)) (8 (7) (10))

```