

	Отчёт по лабораторной работе №23 по курсу 1
	студента группы М8О-215Бв-24 Ярусов Ярослав , № по списку 31
	Адреса www, e-mail, jabber, skype iaroslav
	Работа выполнена: “20“ сентября 2025г.
	Преподаватель: Речинская Ангелина Юрьевна
	Входной контроль знаний с оценкой
	Отчёт сдан “26“ сентября 2025 г., итоговая оценка
Подпись преподавателя	

☐ **Тема:** программа на языке Си

☐ **Цель работы:** научиться реализовывать деревья выражений

☐ **Задание (вариант 29):** определить число листьев двоичного дерева

☐ **Оборудование (лабораторное):**

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ МБ

НМД _____ ГБ. Терминал _____ адрес _____. Принтер _____

Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор **AMD Ryzen 7 4800U**, ОП **16** ГБ, НМД **1** ТБ. Монитор _____

Другие устройства _____

☐ **Программное обеспечение (лабораторное):**

Операционная система семейства _____, наименование _____ версия _____

Интерпретатор команд _____ версия _____

Система программирования _____ версия _____

Редактор текстов _____ версия _____

Утилиты операционной системы _____

Прикладные системы и программы _____

Местонахождения и имена файлов программ и данных _____

Программное обеспечение ЭВМ студента, если использовалось:

Операционная система семейства _____, наименование _____ версия _____

Интерпретатор команд _____ версия _____

Система программирования _____ версия _____

Редактор текстов _____ версия _____

Утилиты операционной системы _____

Прикладные системы и программы _____

Местонахождения и имена файлов программ и данных _____

- **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями)

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

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

Node* insert(Node* t, int v) {
    if (t == NULL) {
        Node* n = malloc(sizeof(Node));
        n->data = v;
        n->left = NULL;
        n->right = NULL;
        return n;
    }
    if (v < t->data) {
        t->left = insert(t->left, v);
    } else if (v > t->data) {
        t->right = insert(t->right, v);
    }
    return t;
}

Node* del(Node* t, int v) {
    if (t == NULL) return t;

    if (v < t->data) {
        t->left = del(t->left, v);
    } else if (v > t->data) {
        t->right = del(t->right, v);
    } else {
        if (t->left == NULL) {
            Node* r = t->right;
            free(t);
            return r;
        }
        if (t->right == NULL) {
            Node* r = t->left;
            free(t);
            return r;
        }
        Node* tmp = t->right;
        while (tmp->left != NULL) tmp = tmp->left;
        t->data = tmp->data;
        t->right = del(t->right, tmp->data);
    }
    return t;
}

void print(Node* t, int l) {
    if (t == NULL) return;
    for (int i = 0; i < l; i++) printf(" ");
    printf("%d\n", t->data);
    print(t->left, l + 1);
    print(t->right, l + 1);
}
```

```

int leaves(Node* t) {
    if (t == NULL) return 0;
    if (t->left == NULL && t->right == NULL) return 1;
    return leaves(t->left) + leaves(t->right);
}

int main() {
    Node* root = NULL;
    char c;
    int v;

    //a - add, b - del, c - print, d - count, f - finish

    while (1) {
        scanf_s(" %c", &c, 1);
        if (c == 'a') {
            scanf_s("%d", &v);
            root = insert(root, v);
        } else if (c == 'b') {
            scanf_s("%d", &v);
            root = del(root, v);
        } else if (c == 'c') {
            print(root, 0);
        } else if (c == 'd') {
            printf("%d\n", leaves(root));
        } else if (c == 'f') {
            break;
        }
    }

    return 0;
}

```

Пункты 1-7 отчёта составляются **строго до** начала лабораторной работы.

Допущен к выполнению работы. Подпись преподавателя _____

- ☐ **Распечатка протокола** (подклеить листинг окончательного варианта программы с текстовыми примерами, подписанный преподавателем)
- ☐ **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

--	--	--	--	--	--	--

☐ Замечание автора по существу работы _____

☐ Выводы : реализовал дерево выражений _____

☐ Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом _____

Подпись студента _____