**Отчет по лабораторной работе № 23**

по курсу Практикум программирования

Студент группы М8О-111Б-23 Бугренков Владимир Петрович,

№ по списку 4, Контакты e-mail: vladimir.bugrenkov@yandex.ru

Работа выполнена: «27» апреля 2023 г.

Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Отчет сдан « » \_\_\_\_\_\_\_\_\_20 \_\_\_ г., итоговая оценка \_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Тема:** динамические структуры данных, обработка деревьев.
2. **Цель работы:** составить программу на языке Си для построения и обработки дерева двоичного вида, содрежащего узлы типа int
3. **Задание** (*вариант № 7*)**:** Проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона
4. **Оборудование** (лабораторное):

ЭВМ \_\_\_\_\_\_\_\_\_, процессор \_\_\_\_\_\_\_. имя узла сети\_\_\_\_\_\_\_с ОП\_\_\_\_\_ Мб, НМД \_\_\_\_ Мб. Терминал \_\_\_\_\_\_\_\_адрес \_\_\_\_\_\_\_\_. Принтер\_\_\_\_\_\_\_

Другие устройства

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор \_AMD Ryzen 5 \_ с ОП 16 Гб НМД \_\_\_512 Гб. Монитор 1920x1080~60Hz Другие устройства

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Программное обеспечение** (лабораторное):

Операционная система семейства \_\_\_\_\_\_\_\_\_\_\_, наименование \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия

\_\_\_\_\_\_\_\_\_\_\_\_\_

интерпретатор команд \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_

Система программирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Редактор текстов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утилиты операционной системы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства \_Linux\_, наименование \_Ubuntu\_ версия 22.04.2 интерпретатор команд \_GNU bash\_ версия 5.1.16.

Система программирования С.

Редактор текстов emacs версия 29.1

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы Emacs

Местонахождение и имена файлов программ и данных на домашнем компьютере /home/

1. **Идея, метод, алгоритм** решения задачи(в формах:словесной,псевдокода,графической[блок-схема,диаграмма,рисунок,таблица]

Составить программу на языке Си, выполняющую четыре действия: добавление нового узла, визуализация дерева, удаление узла, вычисление некоторой функции от дерева в соответствии с заданным вариантом. Для этого напишем структуру Node и дополнительные функции:

• SearchNode — ищет элемент в дереве

• delete\_knot — удаляет элемент в дереве

• print\_tree — выводит текстовую визуализацию дерева

• add\_knot — добавляет элемент в дерево

• print\_menu — выводит меню

• leaf\_cheak— Проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона

1. **Сценарий выполнения работы** [план работы,первоначальный текст программы в черновике(можно на отдельном листе)итесты либо соображения по тестированию].

Содержание main.c — входная точка программы:

#include **<stdio.h>**#include **<stdbool.h>**#include **<stdlib.h>  
  
typedef struct** Node {  
 **int** elem;  
 **struct** Node \*left;  
 **struct** Node \*right;  
} Node;  
  
Node \*add\_knot(Node \*tree, **int** num);  
  
Node \*delete\_knot(Node \*tree, **int** num);  
  
Node \*find\_Min(Node \*tree);  
  
**int** SearchNode(Node \*tree, **int** number);  
  
**void** print\_tree(Node \*tree, **int** number);  
  
**void** print\_menu();  
  
  
Node \*add\_knot(Node \*tree, **int** num) {  
 **if** (tree == **NULL**) {  
 Node \*p = malloc(**sizeof**(Node));  
 p->elem = num;  
 p->left = p->right = **NULL**;  
 **return** p;  
 } **else if** (num < tree->elem) {  
 tree->left = add\_knot(tree->left, num);  
 } **else if** (num > tree->elem) {  
 tree->right = add\_knot(tree->right, num);  
 } **else** {  
 printf(**"(ㆆ\_ㆆ) Элемент с таким значением уже есть, попробуйте снова с другим элементом. \n"**);  
 **return** tree;  
 }  
 **return** tree;  
}  
  
**void** print\_tree(Node \*tree, **int** number) {  
 **if** (tree != **NULL**) {  
 print\_tree(tree->right, number + 1);  
 **for** (**int** i = 0; i < number; i++) printf(**"\t"**);  
 printf(**"%d\n"**, tree->elem);  
 print\_tree(tree->left, number + 1);  
 }  
}  
  
Node \*delete\_knot(Node \*tree, **int** number) {  
 **if** (tree == **NULL**) {  
 **return NULL**;  
 }  
  
 **if** (number > tree->elem) {  
 tree->right = delete\_knot(tree->right, number);  
 } **else if** (number < tree->elem) {  
 tree->left = delete\_knot(tree->left, number);  
 } **else** {  
 **if** (tree->left == 0 && tree->right == 0) {  
 free(tree);  
 **return NULL**;  
 } **else if** (tree->left == 0 || tree->right == 0) {  
 Node \*temp;  
 **if** (tree->left == 0) {  
 temp = tree->right;  
 } **else** {  
 temp = tree->left;  
 }  
 free(tree);  
 **return** temp;  
 } **else** {  
 Node \*temp;  
 temp = find\_Min(tree->right);  
 tree->elem = temp->elem;  
 tree->right = delete\_knot(tree->right, temp->elem);  
 }  
 }  
 **return** tree;  
}  
  
**void** print\_menu() {  
 printf(**"\n█ Выберите действие из списка, написав его номер: \n"**);  
 printf(**"█ Введите 0, чтобы выйти.\n"**);  
 printf(**"█ Введите 1, чтобы добавить в дерево элемент.\n"**);  
 printf(**"█ Введите 2, чтобы вывести текстовую визуализацию дерева.\n"**);  
 printf(**"█ Введите 3, чтобы удалить из дерева элемент.\n"**);  
 printf(**"█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.\n"**);  
}  
  
  
**int** SearchNode(Node \*tree, **int** number) {  
 **if** (tree == **NULL**) {  
 **return** 0;  
 } **else if** (tree->elem == number) {  
 **return** 1;  
 } **else if** (tree->elem < number) {  
 **if** (tree->right != 0) {  
 **return** SearchNode(tree->right, number);  
 } **else** {  
 **return** 0;  
 }  
 } **else if** (tree->elem > number) {  
 **if** (tree->left != 0) {  
 **return** SearchNode(tree->left, number);  
 } **else** {  
 **return** 0;  
 }  
 }  
}  
  
Node \*find\_Min(Node \*tree) {  
 **if** (tree == **NULL**) {  
 **return NULL**;  
 } **else if** (tree->left != **NULL**) {  
 **return** find\_Min(tree->left);  
 }  
 **return** tree;  
}  
  
  
**int** leaf\_cheak(**const** Node \*tree, **int** start\_range, **int** end\_range) {  
 **int** flag = 1;  
  
 **if** (tree->left == **NULL** && tree->right == **NULL**) {  
  
 **if** (((tree->elem) < start\_range) || (((tree->elem) > end\_range))) {  
 *// Нашелся лист который не подошел, рекурсия вернет при умножении 0* **return** 0;  
 } **else** {  
  
 **return** 1;  
 }  
 }  
  
 **if** (tree->left != **NULL** && tree->right == **NULL**) {  
 flag \*= leaf\_cheak(tree->left, start\_range, end\_range);  
 }  
 **if** (tree->left == **NULL** && tree->right != **NULL**) {  
 flag \*= leaf\_cheak(tree->right, start\_range, end\_range);  
 }  
 **if** (tree->left != **NULL** && tree->right != **NULL**) {  
 flag \*= leaf\_cheak(tree->left, start\_range, end\_range);  
 flag \*= leaf\_cheak(tree->right, start\_range, end\_range);  
 }  
 **return** flag;  
}  
  
  
**int** main() {  
 Node \*tree = **NULL**;  
 print\_menu();  
 **int** menu\_number;  
 scanf(**"%d"**, &menu\_number);  
 **while** (menu\_number != 0) {  
 **int** number;  
 **switch** (menu\_number) {  
 **case** 0:  
 printf(**"◁ Завершение работы... \n"**);  
 exit(0);  
 **case** 1:  
 printf(**"▷ Введите значение элемента: \n"**);  
 scanf(**"%d"**, &number);  
  
 **if** (SearchNode(tree, number) == 1) {  
 printf(**"(ㆆ\_ㆆ) Такой элемент уже содержится в дереве, попробуйте другой. \n"**);  
 } **else** {  
 tree = add\_knot(tree, number);  
 printf(**"◁ Значение элемента успешно добавлено в дерево. \n"**);  
 }  
 **break**;  
 **case** 2:  
 **if** (tree != **NULL**) {  
 printf(**"◁ Полученное дерево: \n"**);  
 print\_tree(tree, 1);  
 } **else** {  
 printf(**"◁ Дерево пустое. \n"**);  
 }  
 **break**;  
 **case** 3:  
 **if** (tree != **NULL**) {  
 printf(**"▷ Введите элемент, который хотите удалить из дерева: \n"**);  
 scanf(**"%d"**, &number);  
  
 **if** (SearchNode(tree, number) == 1) {  
 delete\_knot(tree, number);  
 printf(**"◁ Элемент успешно удален из дерева. \n"**);  
 } **else** {  
 printf(**"(ㆆ\_ㆆ) Такого элемента не содержится в дереве. \n"**);  
 }  
 } **else** {  
 printf(**"(ㆆ\_ㆆ) Дерево пустое, удалять нечего. \n"**);  
 }  
 **break**;  
 **case** 4:  
 **if** (tree != **NULL**) {  
 printf(**"▷ Введите заданный диапозон, следуюзим образом: a;b\n"**);  
  
 **int** start\_range, end\_range;  
 scanf(**"%d;%d"**, &start\_range, &end\_range);  
 printf(**"\n"**);  
 **if** (leaf\_cheak(tree, start\_range, end\_range)) {  
 printf(**"Во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел [%d;%d]! \n"**,  
 start\_range, end\_range);  
 } **else** {  
 printf(**"НЕ во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел![%d;%d]! \n"**,  
 start\_range, end\_range);  
 }  
 } **else** {  
 printf(**"◁ Дерево пустое. \n"**);  
 }  
  
  
 **break**;  
 **default**:  
 printf(**"(ㆆ\_ㆆ) Неправильный ввод, попробуйте еще раз: \n"**);  
 }  
  
 print\_menu();  
 scanf(**"%d"**, &menu\_number);  
 }  
 **return** 0;  
}

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем).

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Desktop/Мои материалы/Информатика/2 Семестр/ЛР23\_$ gcc main.c

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Desktop/Мои материалы/Информатика/2 Семестр/ЛР23\_$ ./a.out

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Дерево пустое.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

(ㆆ\_ㆆ) Дерево пустое, удалять нечего.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

5

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

4

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

6

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

6

5

4

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

4

▷ Введите заданный диапозон, следуюзим образом: a;b

4;6

Во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел [4;6]!

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

4

▷ Введите заданный диапозон, следуюзим образом: a;b

2;3

НЕ во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел![2;3]!

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

4

▷ Введите заданный диапозон, следуюзим образом: a;b

2;4

НЕ во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел![2;4]!

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

4

▷ Введите заданный диапозон, следуюзим образом: a;b

2;5

НЕ во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел![2;5]!

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

▷ Введите элемент, который хотите удалить из дерева:

6

◁ Элемент успешно удален из дерева.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

5

4

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

▷ Введите элемент, который хотите удалить из дерева:

2342342

(ㆆ\_ㆆ) Такого элемента не содержится в дереве.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

0

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Desktop/Мои материалы/Информатика/2 Семестр/ЛР23\_$ gcc main.c

qwental@DESKTOP-NKF1EUK:/mnt/c/Users/Holiday/Desktop/Мои материалы/Информатика/2 Семестр/ЛР23\_$ ./a.out

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

6

(ㆆ\_ㆆ) Неправильный ввод, попробуйте еще раз:

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

8

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

4

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

12

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

12

8

4

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

2

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

6

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

10

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

14

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

14

12

10

8

6

4

2

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

8

(ㆆ\_ㆆ) Такой элемент уже содержится в дереве, попробуйте другой.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

4

▷ Введите заданный диапозон, следуюзим образом: a;b

2;6

НЕ во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел![2;6]!

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

4

▷ Введите заданный диапозон, следуюзим образом: a;b

2;15

Во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел [2;15]!

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

1

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

3

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

5

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

7

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

9

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

11

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

13

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

15

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

4

▷ Введите заданный диапозон, следуюзим образом: a;b

0;0

НЕ во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел![0;0]!

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

4

▷ Введите заданный диапозон, следуюзим образом: a;b

1;15

Во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел [1;15]!

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

4

▷ Введите заданный диапозон, следуюзим образом: a;b

4;15

НЕ во всех листьях двоичного дерева элементы находятся в заданном диапозоне чисел![4;15]!

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

▷ Введите элемент, который хотите удалить из дерева:

15

◁ Элемент успешно удален из дерева.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

▷ Введите элемент, который хотите удалить из дерева:

14

◁ Элемент успешно удален из дерева.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

13

12

11

10

9

8

7

6

5

4

3

2

1

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

▷ Введите элемент, который хотите удалить из дерева:

13

◁ Элемент успешно удален из дерева.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

12

11

10

9

8

7

6

5

4

3

2

1

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

▷ Введите элемент, который хотите удалить из дерева:

10

◁ Элемент успешно удален из дерева.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

12

11

9

8

7

6

5

4

3

2

1

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

▷ Введите элемент, который хотите удалить из дерева:

11

◁ Элемент успешно удален из дерева.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

12

9

8

7

6

5

4

3

2

1

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

1

▷ Введите значение элемента:

10

◁ Значение элемента успешно добавлено в дерево.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

12

10

9

8

7

6

5

4

3

2

1

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

▷ Введите элемент, который хотите удалить из дерева:

9

◁ Элемент успешно удален из дерева.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

12

10

8

7

6

5

4

3

2

1

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

3

▷ Введите элемент, который хотите удалить из дерева:

4

◁ Элемент успешно удален из дерева.

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

2

◁ Полученное дерево:

12

10

8

7

6

5

3

2

1

█ Выберите действие из списка, написав его номер:

█ Введите 0, чтобы выйти.

█ Введите 1, чтобы добавить в дерево элемент.

█ Введите 2, чтобы вывести текстовую визуализацию дерева.

█ Введите 3, чтобы удалить из дерева элемент.

█ Введите 4, чтобы проверить, находятся ли во всех листьях двоичного дерева элементы со значениями из заданного диапазона.

0

**9 Дневник отладки** должен содержать дату и время сеансов отладки и основные события(ошибки в сценарии и программе,нестандартные ситуации) 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 |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Выводы**

Я научился работать с двоичными деревьями в си.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_