

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ
(ТУСУР)

**Кафедра автоматизированных систем управления
(АСУ)**

**Отчет по лабораторной работе № 1 по дисциплине
«Структуры и алгоритмы обработки данных в ЭВМ»**

Выполнил

: Небиев Н.И.

« 26 » июля 20 23 г.

Проверил: _____

« _____ » _____ 20 _____ г.

СОДЕРЖАНИЕ

1. Тема работы.
2. Цель работы.
3. Индивидуальное задание.
4. Результаты работы программы.
5. Выводы.

Приложение А. Листинг программы.

1. Тема работы.
«Бинарные деревья».

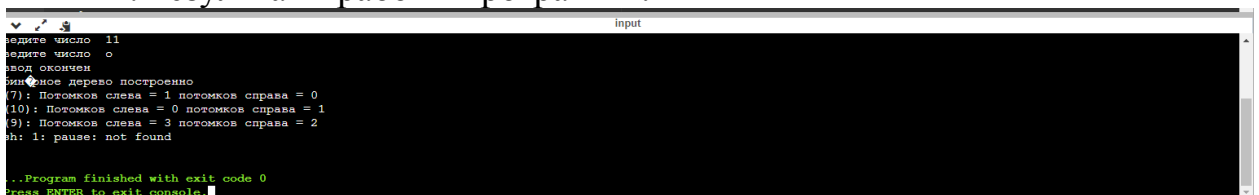
2. Цель работы.

Получить практические навыки представления в памяти ЭВМ структуры данных «бинарное дерево», реализовать на языке программирования C/C++ алгоритмы работы с деревьями.

3. Индивидуальное задание.

Дана последовательность чисел. Написать программу формирования и вывода бинарного дерева на экран. Найти в построенном бинарном дереве все вершины, для которых выполняется условие: количество потомков в левом поддереве отличается от количества потомков в правом поддереве на 1. После выполнения программы очистить память, занятую древовидной структурой

4. Результаты работы программы.



```
input
Введите число 11
Введите число 0
Нажмите Enter для окончания
Бинарное дерево построено
(7): Потомков слева = 1 потомков справа = 0
(10): Потомков слева = 0 потомков справа = 1
(9): Потомков слева = 3 потомков справа = 2
Пауза: 1: pause: not found
..Program finished with exit code 0
Press ENTER to exit console.
```

5. Выводы.

В результате выполнения лабораторной работы получены практические навыки представления в памяти ЭВМ структуры данных «бинарное дерево», реализованы на языке программирования C/C++ алгоритмы работы с деревьями.

Приложение А

```
#include <iostream>

using namespace std;

struct node
{
    int value;           //значение вершины
    node *L, *R;       //Левая и Правая часть дерева
};

void add(node **n, int aData)
{
    if (*n == NULL)
    {
        *n = new node;    //Выделяем память, создаем новую ветку
        (*n)->value = aData; //Кладем в выделенное место аргумент
        (*n)->L = NULL;
        (*n)->R = NULL;    //Очищаем память для следующего роста
        return;
    }
    if (aData > (*n)->value )
    {
        add( &(*n)->R, aData );
    }
    if (aData <= (*n)->value)
    {
        add( &(*n)->L, aData);
    }
};

int TreeCalc(node **n, int allChild) {
    int ChLeft, ChRigth;
    if (*n == NULL) //дойдя до dna(низа древа).
    {
        return allChild = 0; //потомков нет.
    }
    else           //Если узел существует.

//Подсчёт количества потомков для текущего узла.
    ChLeft = TreeCalc( &(*n)->L, allChild); //Для левой ветви.
```

```

ChRigth = TreeCalc( &(*n)->R, allChild); //Для правой ветви.

if ( (ChLeft-ChRigth==1 )||(ChRigth-ChLeft == 1) )
{
    cout<<"("<< (**n).value<<"): Потомков слева = "<< ChLeft<<" потомков
справа = "<< ChRigth<<endl;
}

//потомки текущей вершины = 1 + потомки слева + потомки справа
return allChild = 1 + ChLeft + ChRigth;
};

void freeMem(node **n)
{
    if (*n == NULL) {
        return;
    }
    freeMem(&(*n)->L);
    freeMem(&(*n)->R);
    free(*n);
    *n = NULL;
};

int main()
{
    setlocale(LC_ALL, "rus");
    node *tree = NULL; //Создаем пустое дерево
    cout << "hello" << endl;
    int s; //Число, передаваемое в дерево
    cout << "Для окончание ввода введите любой символ" << endl;
    do
    {
        cout << "ведите число ";
        cin >> s; //Считываем элемент за элементом
        if (cin) {
            add(&tree, s);
        }
        else {
            cout<<"ввод окончен"<<endl;
        }
    } while (cin);
    cout << "бинарное дерево построено" << endl;
    TreeCalc(&tree, 0);
    //сюда добавить вывод сообщение если условие не выполнялось
    freeMem(&tree);
    system("pause");
}

```

```
    return 0;  
}
```