Пензенский государственный университет

Кафедра «Вычислительная техника»

**Отчёт**

По лабораторной работе № 4

По дисциплине: «[Логика и основы алгоритмизации в инженерных задачах](http://moodle.pnzgu.ru/course/view.php?id=48560)»

на тему: «Бинарное дерево поиска»

*Выполнил студент группы 21ВВ2:*

*.*

*Приняли:*

*Митрохин М.А.*

*Юрова О.А.*

Пенза 2022

**Цель работы:**

Изучение организации алгоритмов построения бинарных деревьев.

**Лабораторные задания:**

1. Реализовать алгоритм поиска вводимого с клавиатуры значения в уже созданном дереве.
2. Реализовать функцию подсчёта числа вхождений заданного элемента в дерево.

**Листинг:**

#include <memory.h>

#include <stdlib.h>

#include <locale.h>

#include <time.h>

struct Node {

int data;

struct Node\* left;

struct Node\* right;

};

struct Node\* CreateTree(struct Node\* root, struct Node\* r, int data)

{

if (r == NULL)

{

r = (struct Node\*)malloc(sizeof(struct Node));

if (r == NULL)

{

printf("Ошибка выделения памяти");

exit(0);

}

r->left = NULL;

r->right = NULL;

r->data = data;

if (root == NULL)

return r;

if (data > root->data)

root->right = r;

else

root->left = r;

return r;

}

if (data > r->data)

CreateTree(r, r->right, data);

else

CreateTree(r, r->left, data);

return root;

}

void print\_tree(struct Node\* r, int l)

{

if (r == NULL)

return;

print\_tree(r->right, l + 1);

for (int i = 0; i < l; i++)

printf(" ");

printf("%d\n", r->data);

print\_tree(r->left, l + 1);

}

void search\_elem(struct Node\* r, int data) //поиск по заданному элементу

{

if (r == NULL)

return;

if (r->data < data)

{

search\_elem(r->right, data);

}

else if (r->data > data)

{

search\_elem(r->left, data);

}

if (r->data == data)

printf("%d\n", r->data);

}

int counter(Node\* root, int E) //счётчик

{

if (root == 0)

return 0;

return (root->data == E) + counter(root->left, E) + counter(root->right, E);

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "Russian");

srand(time(NULL));

int D, number;

struct Node\* root = NULL;

printf("Введите размер дерева: ");

scanf("%d", &D);

while (D)

{

int num;

scanf("%d", &num);

root = CreateTree(root, root, num);

D--;

}

printf("Построение дерева окончено\n\n");

print\_tree(root, 0);

printf("\nВведите элемент для поиска и подсчёта: ");

scanf("%d", &D);

search\_elem(root, D);

number = counter(root, D);

if (number)

printf("Кол-во искомых элементов: %d\n", number);

else

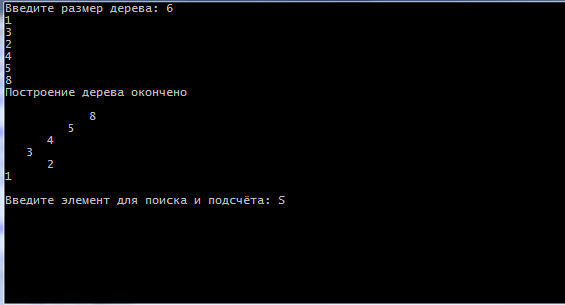
printf("Ничего не найдено\n");

system("pause");

return 0;

}

**Работа программы**

****

**Вывод:** изучил организацию алгоритмов построения бинарных деревьев.