Пензенский государственный университет

Факультет вычислительной техники

Кафедра «Вычислительная техника»

**Отчет**

по лабораторной работе №4

по дисциплине: «Логика и основы алгоритмизации в инженерных задачах»

на тему: «Бинарное дерево поиска»

**Выполнил студент группы 23ВВВ1:**

Евстифеев Д.А

**Приняли:**

Митрохин М. А.

Юрова О. В.

**Название**

Бинарное дерево поиска

**Цель работы**

Выполнить поставленные задачи в соответствии с требованиями

**Лабораторное задание**

**Задание 1:**

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

**Ход работы**

**Листинг А.**

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

// узел бинарного дерева

struct Node {

int data;

struct Node\* left;

struct Node\* right;

} ;

struct Node\* root = NULL;

struct Node\* CreateTree(struct Node\* root, struct Node\* r, int data) { // функция создания дерева

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->left = r;

else root->right = r;

return r;

}

if (data > r->data)

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

else

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

return root;

}

void print\_tree(struct Node\* r, int l) // функция вывода дерева на экран

{

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);

}

struct Node\* find(struct Node\* root, int data) { // фукнция для поиска элемента

if (root->data == data) return root;

if (root->data > data && root->right != NULL) find(root->right, data);

else if (root->left != NULL) find(root->left, data);

else return NULL;

}

int count(struct Node\* root, int data, int cnt) { // фукнция для подсчета количества элемента

if (root->data == data) cnt++;

if (root->data >= data && root->right != NULL) cnt = count(root->right, data, cnt);

else if (root->left != NULL) cnt = count(root->left, data, cnt);

else return cnt;

}

int main()

{

setlocale(LC\_ALL, "Rus");

int D, number, start = 1;

printf("-1 - окончание построения дерева\n");

while (start)

{

printf("Введите число: ");

scanf("%d", &D);

if (D == -1)

{

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

start = 0;

}

else

root = CreateTree(root, root, D);

}

print\_tree(root, 0);

printf("Введи значение number\n");

scanf("%d", &number);

int cnt;

cnt = count(root, number, 0);

printf("%d раз(а)", cnt);

struct Node\* r = find(root,number);

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

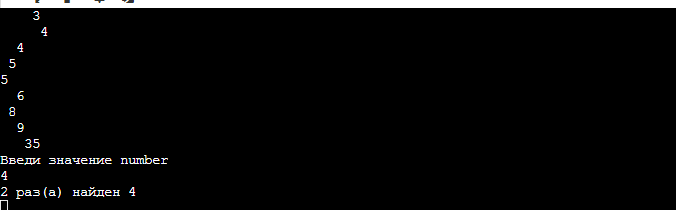
else printf("элемент не найден\n");

scanf("%d", &D);

return 0;

}

Результат программы:



**Вывод:** в ходе выполнения лабораторной работы было построено бинарное дерево поиска и выполнены задания.