Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики

и радиоэлектроники»

Специальность «Программная инженерия»

Кафедра инженерной психологии и эргономики

Учебная дисциплина «Основы алгоритмизации и программирования»

ОТЧЕТ

по лабораторной работе №14

«Бинарное дерево»

Вариант 27

Подготовил: Сивцов А.Д.

Проверил: Усенко Ф.В.

Минск 2024

**Цель работы:** сформировать знания и умения по работе с подпрограммами, приобрести навыки написания программ с использованием бинарных деревьев.

Задание: Построить дерево поиска с элементами — символами. Определить число повторяющихся символов в дереве и удалить дубликаты.

#include <iostream>

using namespace std;

struct tnode {

char field;

int count;

struct tnode\* left;

struct tnode\* right;

tnode() {};

tnode(char c) : field(c), count(1), left(nullptr), right(nullptr) {};

};

void treeprintprefix(tnode\* tree) {

if (tree != NULL) {

cout << tree->field;

treeprintprefix(tree->left);

treeprintprefix(tree->right);

}

}

void treeprintinfix(tnode\* tree) {

if (tree != NULL) {

treeprintinfix(tree->left);

cout << tree->field;

treeprintinfix(tree->right);

}

}

void treeprintpostfix(tnode\* tree) {

if (tree != NULL) {

treeprintinfix(tree->left);

treeprintinfix(tree->right);

cout << tree->field;

}

}

struct tnode\* addnode(char x, tnode\* tree) {

if (tree == NULL) {

tree = new tnode(x);

return tree;

}

if (x < tree->field) {

tree->left = addnode(x, tree->left);

}

else if (x > tree->field) {

tree->right = addnode(x, tree->right);

}

else tree->count++;

return tree;

}

void printT(tnode\* tree) {

if (tree == nullptr) {

return;

}

printT(tree->left);

cout << "Количество " << tree->field << ": " << tree->count << endl;

printT(tree->right);

}

void deleteTree(tnode\* tree) {

if (tree != NULL) {

deleteTree(tree->left);

deleteTree(tree->right);

delete tree;

}

}

int main() {

setlocale(LC\_ALL, "Russian");

tnode\* root = nullptr;

int size;

char element;

cout << "Введите количество символов:" << endl;

cin >> size;

int i = 0;

if (size <= 0) {

cout << "Ошибка! Введено неверное количество символов." << endl;

return 0;

}

while (i < size) {

cout << "Введите символ: " << endl;

cin >> element;

root = addnode(element, root);

i++;

}

cout << "Частота вхождения символов..." << endl;

printT(root);

return 0;

}

Результат работы программы представлен на рисунке 1

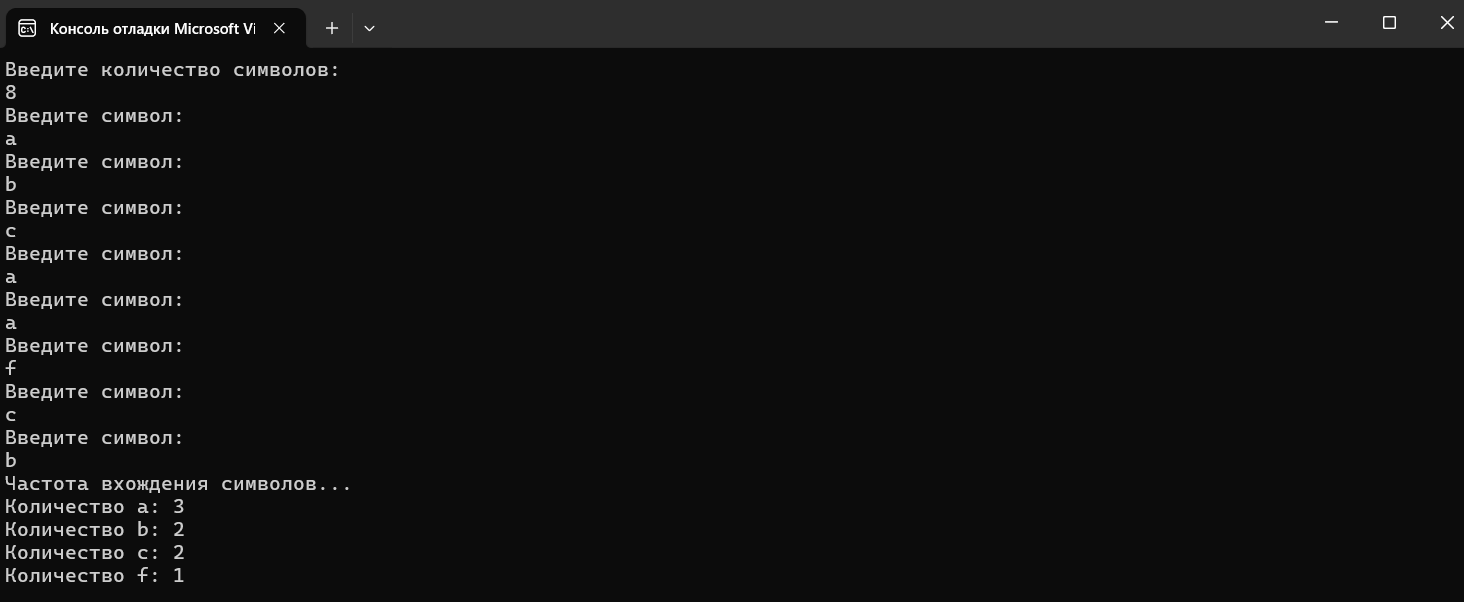


Рисунок 1 – Результат работы программы

Блок-схема работы программы представлена на рисунке 2.

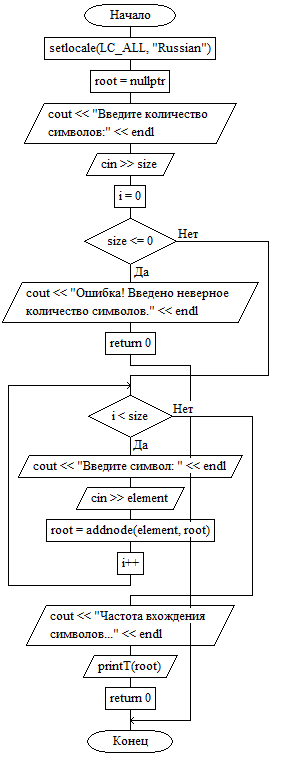


Рисунок 2

**Вывод:** в ходе выполнения работы была достигнута цель данной лабораторной работы: сформировать знания и умения по работе с подпрограммами, приобрести навыки написания программ с использованием бинарных деревьев.