*Національний університет харчових технологій*

*Інформаційних систем*

**Лабораторна робота №19**

з дисципліни *Основи програмування та алгоритмічні мови*

на тему: *Розв’язування задач з використанням динамічних дерев*

Студент *1* курсу *2* групи

Спеціальності *122 «Комп’ютерні науки»*

*Держій Д.Ю*

Варіант завдання 4

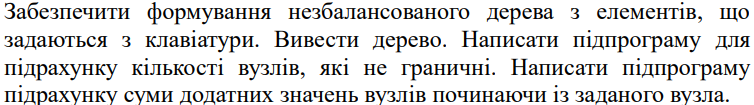
Викладач *доцент к.т.н.*

*Грибков С.В*

Дата здачі *16.10.2022*

Оцінка

Київ-2022

Завдання 1:  


Код програми:

#include <time.h>

#include <stdlib.h>

#include <assert.h>

#include <stdio.h>

#include <windows.h>

#include <iostream>

using namespace std;

struct node

{

int x;

node\* l;

node\* r;

};

void show(node\* mt)

{

if (mt != nullptr)

{

cout << "Вузол " << mt->x << ";\n";

show(mt->l);

show(mt->r);

}

}

int count(node\* mt)

{

if (mt == nullptr || (mt->l == nullptr && mt->r == nullptr))

return 0;

return 1 + count(mt->l) + count(mt->r);

}

int value(node\* mt)

{

if (mt == nullptr)

return 0;

int sum = 0;

if (mt->x > 0)

sum += mt->x;

sum += value(mt->l);

sum += value(mt->r);

return sum;

}

void add(int x, node\*& mt)

{

if (mt == nullptr)

{

mt = new node;

mt->x = x;

mt->l = nullptr;

mt->r = nullptr;

}

else

{

if (count(mt->l) < count(mt->r))

{

if (mt->l != nullptr)

add(x, mt->l);

else

{

mt->l = new node;

mt->l->l = nullptr;

mt->l->r = nullptr;

mt->l->x = x;

}

}

else

{

if (mt->r != nullptr)

add(x, mt->r);

else

{

mt->r = new node;

mt->r->l = nullptr;

mt->r->r = nullptr;

mt->r->x = x;

}

}

}

}

int main()

{

setlocale(LC\_CTYPE, "ukr");

const int n = 15;

int i;

int el[n];

cout << "Введіть " << n << " елементів для дерева: ";

for (i = 0; i < n; i++)

{

cin >> el[i];

}

node\* tree = nullptr;

cout << "\nБудуємо дерево\n";

i = 0;

while (i < n)

{

add(el[i], tree);

i++;

}

cout << "\nЗгенеровані вузли дерева:\n";

show(tree);

int noleaf = count(tree);

cout << "Кількість не граничних вузлів дерева: " << noleaf << endl;

int sum = value(tree);

cout << "Сума додатних значень вузлів: " << sum << endl;

delete tree;

return 0;

}

Результат виконаної роботи  
