Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №14

за 2 семестр

По дисциплине: «ОАиП»

Тема: «Динамические структуры данных: списки и деревья»

Выполнил:

Студент 1 курса

Группы АС-59

Левоцкий Н.

Цыганов Е.

Проверил:

Гирель Т.Н.

2021

Лабораторная работа №14

Динамические структуры данных: списки и деревья

Цель работы: приобретение навыков работы с динамической памятью и указателями на С/C++; изучение принципов работы с динамическими структурами данных: списками и деревьями.

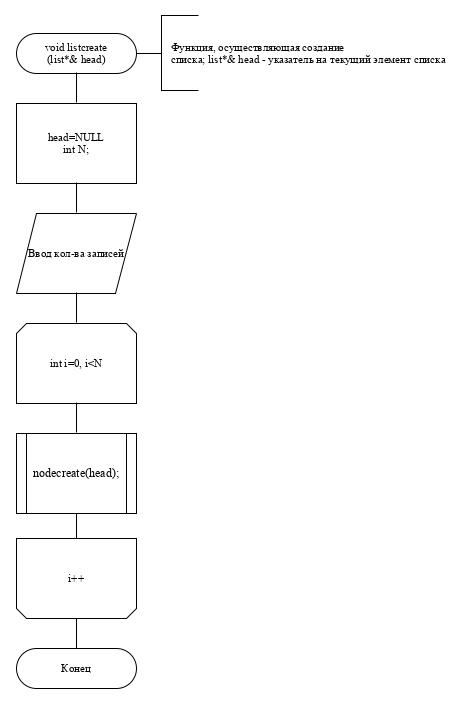
Вариант 1.5

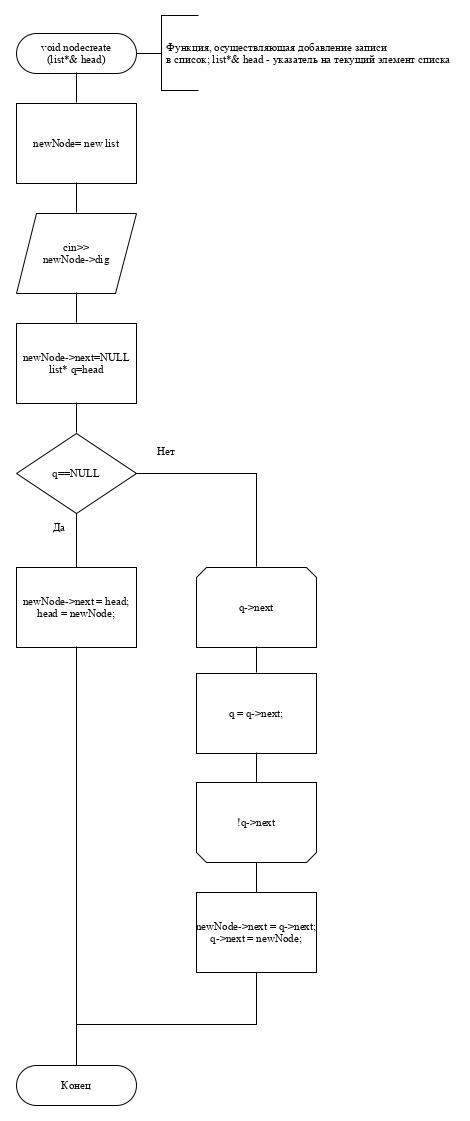
**Задание:**

Общие требования: в начале программы вывести задание; в процессе работы выводить подсказки пользователю (что ему нужно ввести, чтобы продолжить выполнение программы). Основные алгоритмы, ввод/вывод списков реализовать в виде функций с необходимыми параметрами. После работы программы вся динамически выделенная память должна быть освобождена.

Написать программу, которая вводит с клавиатуры список целых чисел, строит инверсную копию списка, затем вводит с клавиатуры сбалансированное дерево и считает количество листьев дерева.

**Блок-схема:**

****

****

**Текст программы:**

#include<Windows.h>

#include <iostream>

using namespace std;

struct list

{

int dig; // число

list\* next; // указатель на следующий элемент

};

struct node {

int info;

node\* l, \* r;

};

node\* tree = NULL;

int Count(node\* Root, int& count);

void delete\_list(list\* head); //удаление списка

void listcreate(list\*& head);

void nodecreate(list\*& head);

void print\_reversed(list\* node);

void push(int a, node\*\* t); //запись элемента в дерево

int empty(list\* head); //проверка на пустоту списка

void print(node\* t, int u,int &count); //вывод дерева

void delete\_tree(node\*& t);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

cout << "Написать программу, которая вводит с клавиатуры список целых чисел,\nстроит инверсную копию списка,\nзатем вводит с клавиатуры сбалансированное дерево и считает количество листьев дерева." << endl;

list\* head = NULL;

listcreate(head);

cout << endl<<"Инверсный список:\n";

print\_reversed(head);

delete\_list(head);

int n; //Количество элементов

int s;//Число, переваемое в дерево

int count = 0;//колво листьев

cout << "Введите количество элементов дерева: ";

cin >> n;

cout << "Введите " << n << " элементов дерева: " << endl;

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

cin >> s;

push(s, &tree);

}

cout << "Ваше дерево: " << endl;

print(tree, 0,count);

cout << "Количество листьев: " << count<<endl;

delete\_tree(tree);

}

void nodecreate(list\*& head)

{

list\* newNode = new list;

cin >> newNode->dig;

newNode->next = NULL;

list\* q = head;

if (q == NULL)

{

newNode->next = head;

head = newNode;

}

else

{

while (q->next)

{

q = q->next;

}

newNode->next = q->next;

q->next = newNode;

}

}

void print\_reversed(list\* node)

{

if (node != NULL)

{

print\_reversed(node->next);

printf("%d ", node->dig);

cout << endl;

}

}

void listcreate(list\*& head)

{

head = NULL;

int N;

cout << "Введите количество записей: ";

cin >> N;

cout << "Вводите числа: " << endl;

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

{

nodecreate(head);

}

}

int empty(list\* head)

{

if (head == NULL) {

return 0;

}

else {

return 1;

}

}

void delete\_list(list\* head)

{

if (empty(head) == 1) {

if (head) {

delete\_list(head->next);

delete head;

}

}

}

void push(int a, node\*\* t)

{

if ((\*t) == NULL)

{

(\*t) = new node;

(\*t)->info = a;

(\*t)->l = (\*t)->r = NULL;

return;

}

if (a > (\*t)->info) push(a, &(\*t)->r);

else push(a, &(\*t)->l);

}

void print(node\* t, int u, int& count)

{

if (t == NULL) return;

else

{

print(t->l, ++u,count);

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

cout << "|";

cout << t->info << endl;

u--;

}

print(t->r, ++u,count);

if ((t->l == NULL) && (t->r == NULL))

count = count ++;

}

void delete\_tree(node\*& t)

{

if (t != NULL)

{

delete\_tree(t->l);

delete\_tree(t->r);

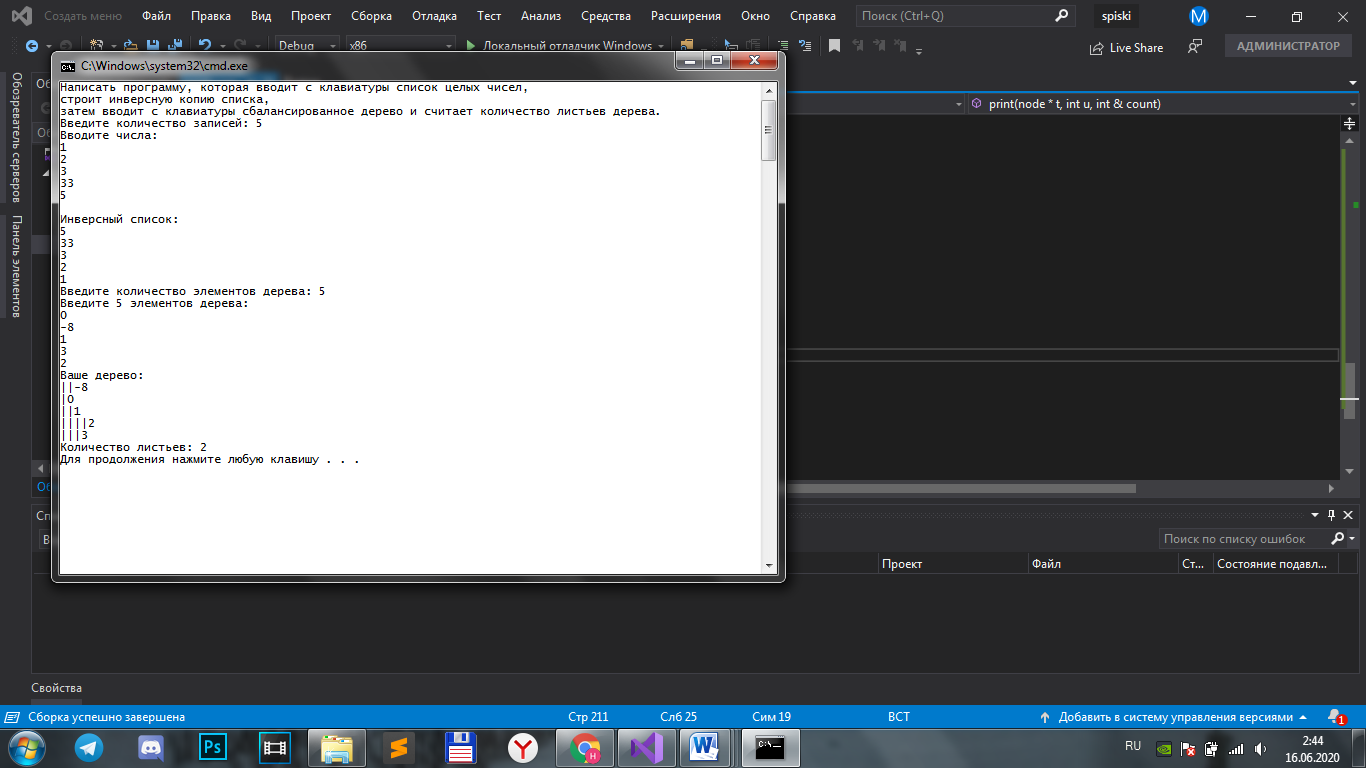
delete t;

t = NULL;

}

}

**Результаты выполнения программы:**



Вывод: приобрел навыки работы с динамической памятью и указателями на С/C++; изучил принципы работы с динамическими структурами данных: списками и деревьями.