МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №14

По дисциплине «Основы алгоритмизации и программирования»

За 2 семестр

Тема: «Динамические структуры: списки и деревья»

Выполнила:

студентка 1 курса

группы АС-56

Карпенко М.В.

Проверил:

Гирель Т.Н.

Брест 2020

*Цель работы:* Приобретение навыков работы с динамической памятью и указателями на C. Изучение принципов работы с динамическими структурами данных: списками и деревьями.

*Задание:*

Написать программу, которая вводит с клавиатуры список целых чисел, подсчитывает среднее арифметическое элементов списка, затем вводит с клавиатуры дерево поиска и считает количество элементов дерева.

*Код программы:*

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <string.h>

typedef struct num

{

int num;

struct num\* next;

} NUM;

typedef struct tree

{

int num;

struct tree\* right;

struct tree\* left;

} TREE;

NUM\* addNumToList(NUM\* head, int num)

{

NUM\* newNum = (NUM\*)malloc(sizeof(NUM));

newNum->next = 0;

newNum->num = num;

if (!head) return newNum;

newNum->next = head;

return newNum;

}

TREE\* addNumToTree(TREE\* head, int num)

{

printf("\nAdd %d\n", num);

TREE\* buff = head;

TREE\* newTree = (TREE\*)malloc(sizeof(TREE));

newTree->right = 0;

newTree->left = 0;

newTree->num = num;

if (!head) return newTree;

while (1)

{

if (newTree->num > head->num)

if (head->right)

head = head->right;

else

{

head->right = newTree;

break;

}

else if (newTree->num < head->num)

if (head->left)

head = head->left;

else

{

head->left = newTree;

break;

}

}

return buff;

}

float getAverageArithmeticFromList(NUM\* head)

{

int i = 0;

int sum = 0;

while (head)

{

printf("%d[%d], ", head->num, i + 1);

sum += head->num;

head = head->next;

i++;

}

return (float)sum / i;

}

int getAmountOfNumsInTree(TREE\* head, int state)

{

static int amount;

if (state) amount = 0;

if (head)

{

amount++;

getAmountOfNumsInTree(head->left, 0);

getAmountOfNumsInTree(head->right, 0);

printf("%d ,", head->num);

}

if (state) return amount;

}

void main()

{

setlocale(0, "Russian");

int num, ch = 1;

NUM\* headOfList = 0;

while (ch)

{

printf("\n Введите число для его добавления в список : ");

scanf("%d", &num);

headOfList = addNumToList(headOfList, num);

printf("\n Хотите продолжить?( Да - 1 / Нет - 0 ) \n ->");

scanf("%d", &ch);

}

printf("\nВведенные числа : ");

printf("\n\tСреднее арифметическое : %.4f\n", getAverageArithmeticFromList(headOfList));

ch = 1;

TREE\* headOfTree = 0;

while (ch)

{

printf("\n Введите число для его добавления в дерево : ");

scanf("%d", &num);

headOfTree = addNumToTree(headOfTree, num);

printf("\n Хотите продолжить?( Да - 1 / Нет - 0 ) \n ->");

scanf("%d", &ch);

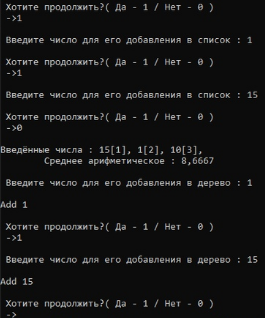
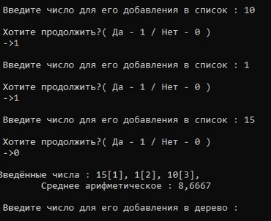
}

printf("\nВведенные в дерево чисел? : ");

printf("\n\tКоличество чисел: %d\n", getAmountOfNumsInTree(headOfTree, 1));

}

*Результаты работы программы:*



*Вывод:* Приобрела навыки работы с динамической памятью и указателями на C. Изучила принципы работы с динамическими структурами данных: списками и деревьями.