**Московский Авиационный Институт**

(Национальный Исследовательский Университет)

Институт №8 «Информационные технологии и прикладная математика»

Кафедра 806 «Вычислительная математика и программирование»

Курсовой проект

по дисциплине «Практикум программирования»

2 семестр

Задание 8.

Тема: «Линейные списки»

|  |  |
| --- | --- |
| Студент: | Доянова В. А. |
| Группа: | М8О-111Б-22 |
| Преподаватель: | Аносова Н. П. |
| Подпись: |  |
| Оценка: |  |

Москва, 2023

Содержание:

**Задание3**

Формулировка3

**Работа4**

Код программы4

Описание алгоритма7

Протокол выполнения программы8

**Вывод9**

**Задание**

**Формулировка**

Составить и отладить программу на языке Си для обработки линейного списка заданной организации с отображением списка на динамические структуры. Навигацию по списку следует реализовать с применением итераторов. Предусмотреть выполнение одного нестандартного и четырех стандартных действий:

1. Печать списка.

2. Вставка нового элемента в список.

3. Удаление элемента из списка.

4. Подсчет длины списка.

Тип элемента списка: строковый.

Вид списка: линейный однонаправленный.

Нестандартное действие: проверить упорядоченность элементов списка.

**Работа**

**Код программы**

head.cpp: #pragma once

#include <string>

using namespace std;

class Node {

public:

string data;

Node\* next;

Node\* prev;

Node() {

data = "";

next = nullptr;

prev = nullptr;

}

Node(string data, Node\* next = nullptr, Node\* prev = nullptr) {

this->data = data;

this->next = next;

this->prev = prev;

}

};

class List {

private:

Node\* first;

Node\* last;

int size;

public:

List();

void read(string filename);

void print();

void insert(string data, int pos);

void remove(int pos);

int getsize();

bool order();

};

functions.cpp: #include <fstream>

#include <iostream>

#include "head.h"

using namespace std;

List::List() {

first = new Node();

last = new Node();

first->next = last;

last->prev = first;

size = 0;

}

void List::read(string filename) {

ifstream file(filename);

if (file.is\_open()) {

string line;

while (getline(file, line)) {

insert(line, size);

}

file.close();

}

}

void List::print() {

Node\* current = first->next;

while (current != last) {

cout << current->data << endl;

current = current->next;

}

}

void List::insert(string data, int pos) {

Node\* current = last->prev;

for (int i = size - 1; i >= pos; i--) {

current = current->prev;

}

Node\* newNode = new Node(data, current->next, current);

current->next = newNode;

newNode->next->prev = newNode;

size++;

}

void List::remove(int pos) {

Node\* current = first;

for (int i = 0; i < pos - 1; i++) {

current = current->next;

}

Node\* nodeToRemove = current->next;

current->next = nodeToRemove->next;

nodeToRemove->next->prev = current;

delete nodeToRemove;

size--;

}

int List::getsize() {

return size;

}

bool List::order() {

Node\* current = first->next;

while (current != last && current->next != last) {

if (current->data.length() > current->next->data.length()) {

return false;

}

current = current->next;

}

return true;

}

main.cpp: #include <iostream>

#include "head.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

List list;

string n;

int a, m;

cout << "Выберите действие: 1. Вывести список;\n\t\t 2. Вставить элемент;\n\t\t 3. Убрать элемент;\n\t\t 4. Вывести размер списка;\n\t\t 5. Проверить упорядоченность списка;\n\t\t 6. Выход\n";

cin >> a;

list.read("input.txt");

while (a != 6) {

if (a == 1) {

cout << "Список: ";

list.print();

}

else if (a == 2) {

cout << "Введите новый элемент и его место в списке: ";

cin >> n >> m;

list.insert(n, m - 1);

cout << "\nСписок: ";

list.print();

}

else if (a == 3) {

cout << "Введите номер элемента, который необходимо удалить: ";

cin >> m;

list.remove(m);

cout << "\nСписок: ";

list.print();

}

else if (a == 4) {

cout << "\nРазмер списка: " << list.getsize() << endl;

}

else {

if (list.order()) {

cout << "Лист упорядочен в порядке возрастания\n";

}

else {

cout << "Лист НЕ упорядочен в порядке возрастания\n";

}

}

cout << "Выберите действие: 1. Вывести список;\n\t\t 2. Вставить элемент;\n\t\t 3. Убрать элемент;\n\t\t 4. Вывести размер списка;\n\t\t 5. Проверить упорядоченность списка;\n\t\t 6. Выход\n";

cin >> a;

}

return 0;

}

**Описание алгоритма**

head.cpp: создаем структуру листа. Объявляем основные переменные (first, last, prev, next, data), которые и составляют лист. Также перечисляем используемые функции.

functions.cpp:

* read – функция считывает строки из файла и заносит их в лист.
* print – функция проходит по листу, начиная с первого элемента, и выводит список.
* insert – в функцию передается значение и желаемое место в списке. Функция находит элемент, который идет перед введенным местом и меняет его позицию, вставляя новый элемент.
* remove – функция работает аналогично функции insert.
* getsize – функция выводит размер списка исходя из количества его элементов.
* order – функция проверяет, упорядочен ли список по возрастанию. Она сравнивает длину каждого элемента и проверяет, сохраняется ли порядок.

main.cpp: пользователь вводит цифру от 1-6, в зависимости от которой происходит вызов определенной функции или завершение программы.

**Протокол выполнения программы**

Входные данные:

12345

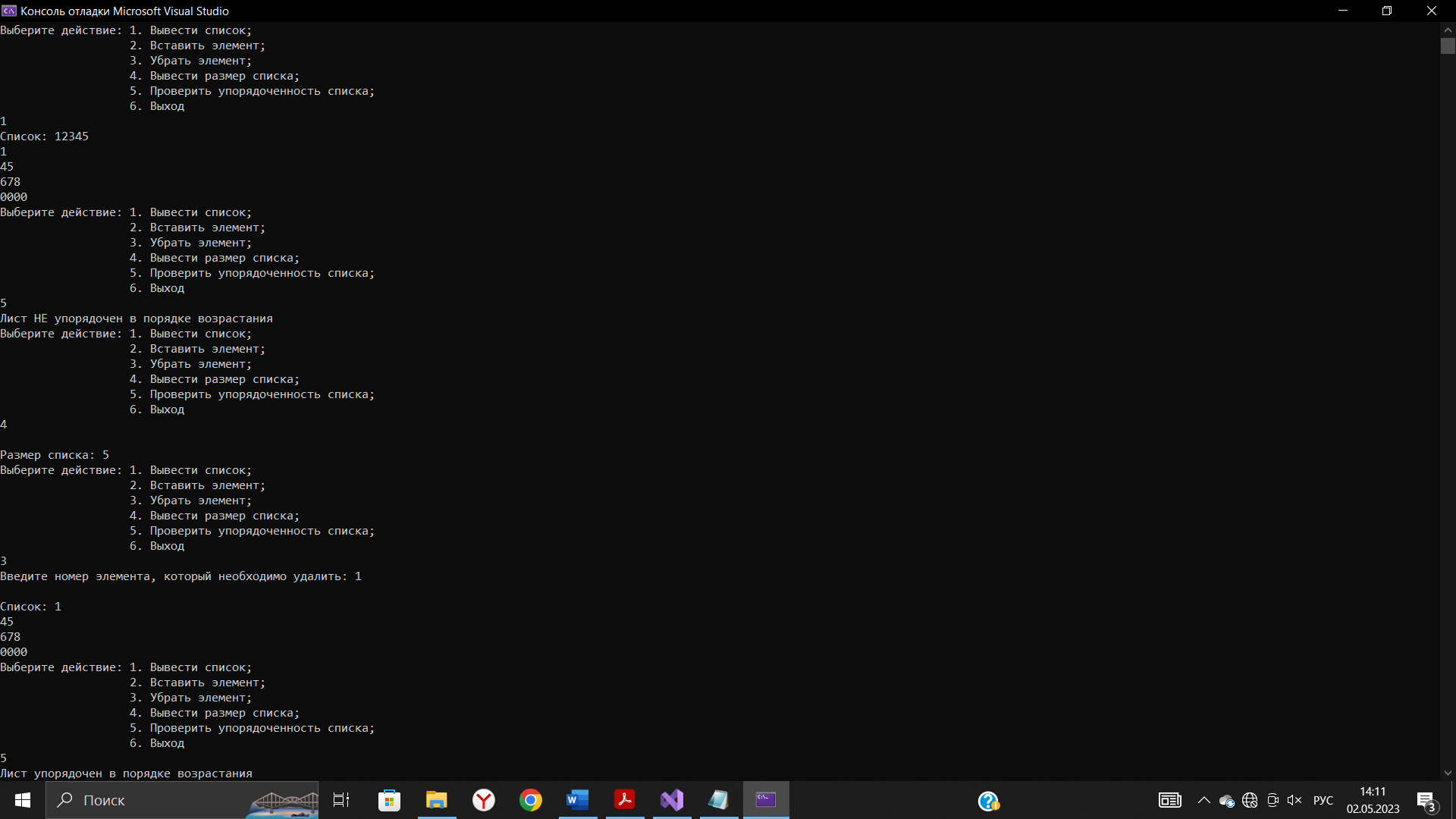
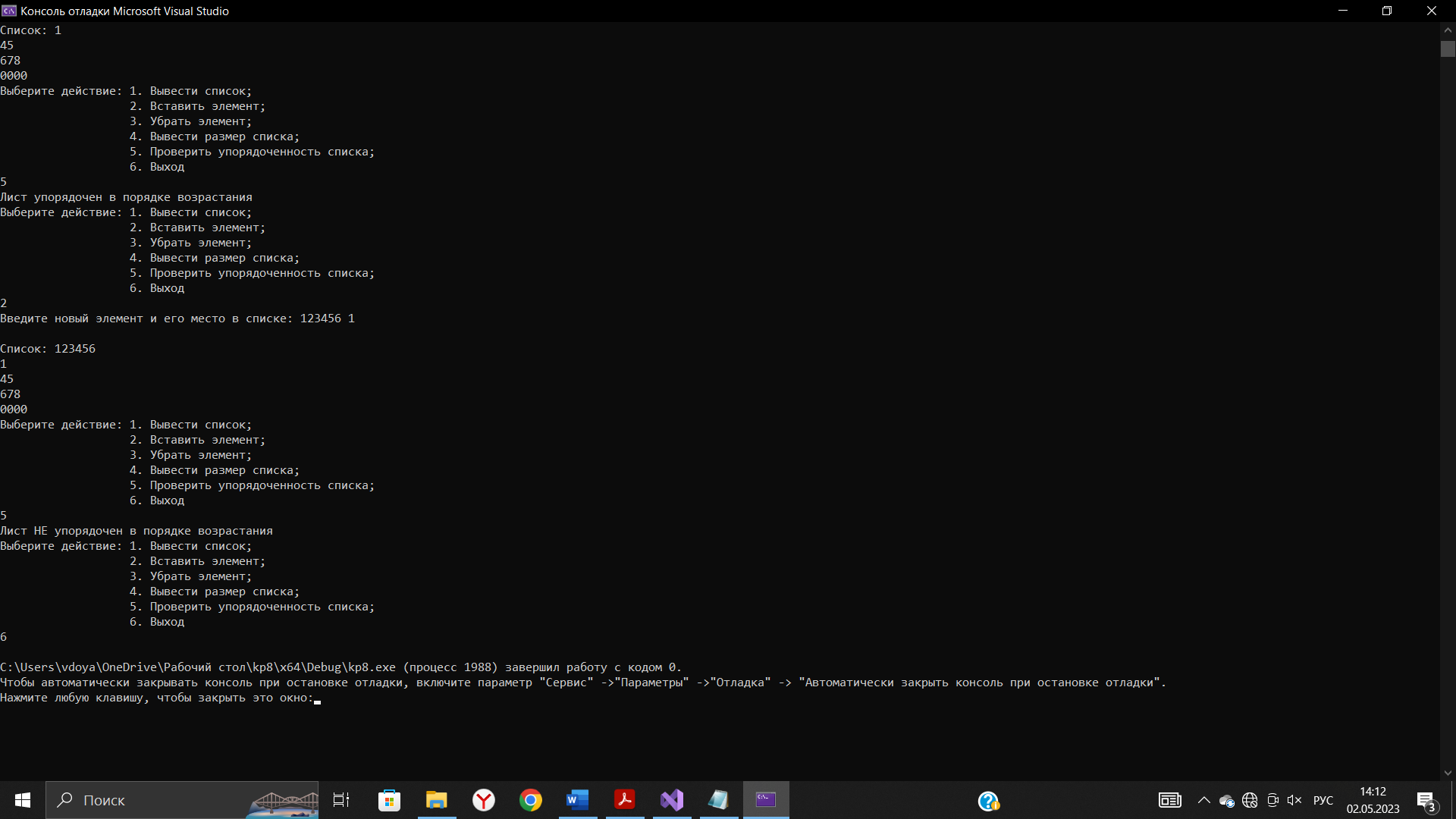
1

45

678

0000

Вывод программы:

** **

**Вывод**

Я научилась работать с линейными списками, осуществлять над ними различные действия. Я улучшила свой навык создания красивого вывода программы, научилась проверять список на упорядоченность.