Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа №6

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамические структуры данных. Односвязные списки »

Выполнил:

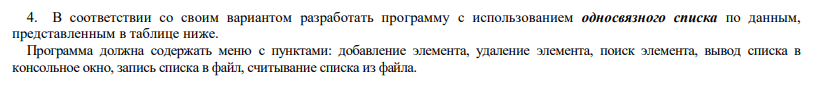
Студент 1 курса 8 группы

Лужецкий Владислав Константинович

Преподаватель: асс. Андронова М.В.

2024, Минск

**//Это условие для всех заданий**



Основной вариант:



#include <fstream>

using namespace std;

struct list

{

int number;

list\* next;

};

void insert(list\*&, float); //функция добавления элемента, передается адрес

float del(list\*&, float); //функция удаления, передается адрес списка и символ,

float find(list\*&, float); //функция поиска

int IsEmpty(list\*); //функция, которая проверяет, пуст ли список

void printList(list\*); //функция вывода

void sumNegativeDoubleDigitElements(list\*); //функция подсчета суммы отрицательных двухзначных элементов

void toFile(list \*&p); //Запись в файл

void fromFile(list \*&p); //Считывание из файла

int variant6()

{

setlocale(LC\_CTYPE, "Russian");

list\* first = NULL;

int choice;

float value;

cout << "Сделайте выбор:" << endl;

cout << " 1 - Ввод числа" << endl;

cout << " 2 - Удаление числа" << endl;

cout << " 3 - Вычисление суммы отрицательных двухзначных элементов" << endl;

cout << " 4 - Запись списка в файл" << endl;

cout << " 5 - Чтение списка из файла" << endl;

cout << " 6 - Поиск числа" << endl;

cout << " 0 - Выход" << endl;

cout << " ? ";

cin >> choice;

while (choice != 0)

{

switch (choice)

{

case 1:

cout << "Введите число "; // добавить число в список

cin >> value;

insert(first, value);

printList(first);

break;

case 2:

if (!IsEmpty(first)) // удалить число из списка

{

cout << "Введите удаляемое число ";

cin >> value;

if (del(first, value))

{

cout << "Удалено число " << value << endl;

printList(first);

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

case 3:

sumNegativeDoubleDigitElements(first); // вычисление суммы отрицательных двухзначных элементов

break;

case 4:

toFile(first);

break;

case 5:

fromFile(first);

break;

case 6:

if (!IsEmpty(first)) // есть ли число в списке?

{

cout << "Введите число ";

cin >> value;

if (find(first, value))

{

cout << "Найдено число: " << value << endl;

printList(first);

}

else

cout << "Число не найдено" << endl;

}

else

cout << "Список пуст" << endl;

break;

default: cout << "Неправильный выбор" << endl;

break;

}

cout << "? ";

cin >> choice;

}

cout << "Конец" << endl;

return 0;

}

void insert(list\*& p, float value) //Добавление числа value в список

{

list\* newP = new list;

if (newP != NULL) //есть ли место?

{

newP->number = value;

newP->next = p;

p = newP;

}

else

cout << "Операция добавления не выполнена" << endl;

}

float del(list \* &p, float value) // Удаление числа

{

list\* previous, \* current, \* temp;

if (value == p->number)

{

temp = p;

p = p->next; // отсоединить узел

delete temp; //освободить отсоединенный узел

return value;

}

else

{

previous = p;

current = p->next;

while (current != NULL && current->number != value)

{

previous = current;

current = current->next; // перейти к следующему

}

if (current != NULL)

{

temp = current;

previous->next = current->next;

delete(temp);

return value;

}

}

return 0;

}

int IsEmpty(list\* p) //Список пустой? (1-да, 0-нет)

{

return p == NULL;

}

void printList(list \* p) //Вывод списка

{

if (p == NULL)

cout << "Список пуст" << endl;

else

{

cout << "Список:" << endl;

while (p != NULL)

{

cout << "-->" << p->number;

p = p->next;

}

cout << "-->NULL" << endl;

}

}

void sumNegativeDoubleDigitElements(list\* p) // Подсчет отрицательных двухзначных элементов

{

float sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > -100 && p->number < 0)

sm = sm + (p->number);

p = p->next;

}

cout << "Сумма отрицательных двухзначных элементов= " << sm << endl;

}

}

void toFile(list\*& p)

{

list\* temp = p;

list buf;

ofstream frm("mList.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

while (temp)

{

buf = \*temp;

frm.write((char\*)&buf, sizeof(list));

temp = temp->next;

}

frm.close();

cout << "Список записан в файл mList.dat\n";

}

void fromFile(list\*& p) //Считывание из файла

{

list buf, \* first = nullptr;

ifstream frm("mList.dat");

if (frm.fail())

{

cout << "\n Ошибка открытия файла";

exit(1);

}

frm.read((char\*)&buf, sizeof(list));

while (!frm.eof())

{

insert(first, buf.number);

cout << "-->" << buf.number;

frm.read((char\*)&buf, sizeof(list));

}

cout << "-->NULL" << endl;

frm.close();

p = first;

cout << "\nСписок считан из файла mList.dat\n";

}

float find(list\*& p, float value) // Поиск

{

list\* previous, \* current, \* temp;

if (value == p->number)

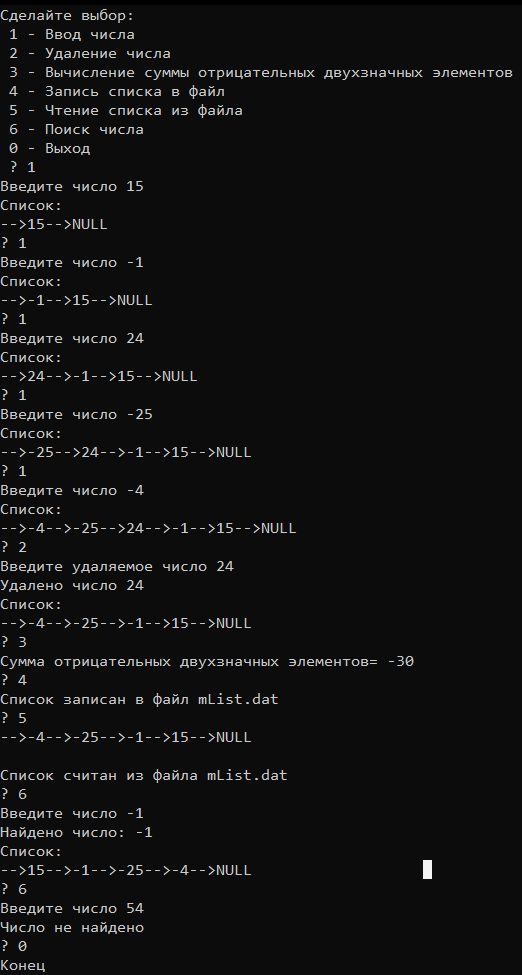
{

return value;

}

return 0;

}



**//На самом деле все задания одинаковые с одинаковыми функциями: ввод\вывод\запись в файл\чтение из файла\поиск\удаление. Задания отличаются лишь функцией, которая у меня находится 3 в меню. Я не вижу смысла скидывать одинаковые функции в допах, там изменятся только комментарии, название -> прототип функции и всё. В допах я буду скидывать код функции и результат реализации этой функции в консольном окне.**

**1 доп:**

****

void sumPositiveElements(list\* p) // Подсчет положительных элементов

{

float sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 0)

sm = sm + (p->number);

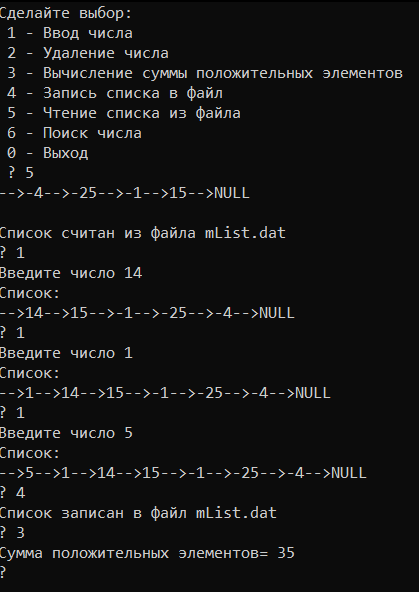
p = p->next;

}

cout << "Сумма положительных элементов= " << sm << endl;

}

}



**2 доп:**

****

void AverageValuePositiveFloatsElements(list\* p)

{

float sm = 0.0;

float av = 0.0;

int counter = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 0.0)

{

sm = sm + (p->number);

counter++;

}

p = p->next;

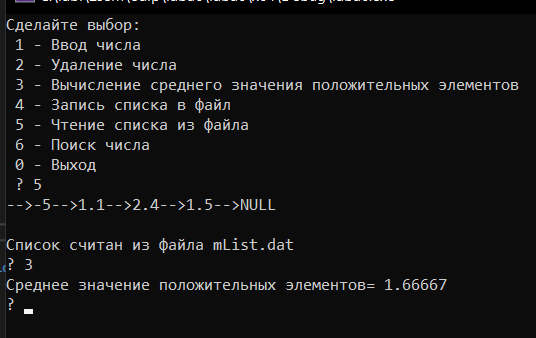
}

av = sm / counter;

cout << "Среднее значение положительных элементов= " << av << endl;

}

}

****

**3 доп:**

****

void sumElemenentDividedFive(list\* p) // Вычисление суммы положительных элементов, кратных 5

{

int sm = 0;

if (p == NULL)

cout << "Список пуст" << endl;

else

{

while (p != NULL)

{

if (p->number > 0 && p->number % 5 == 0)

sm = sm + (p->number);

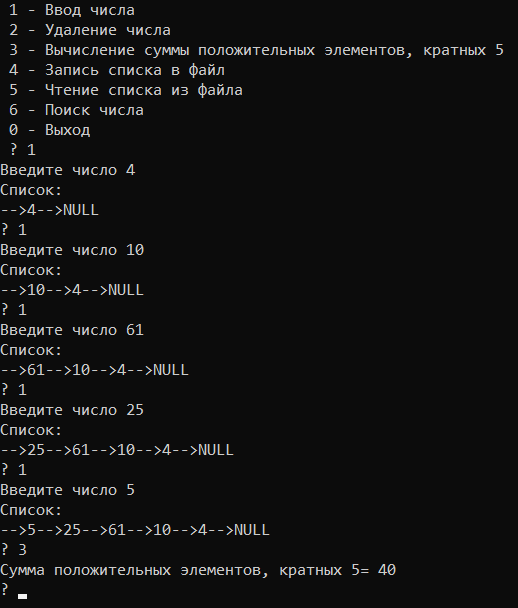
p = p->next;

}

cout << "Сумма положительных элементов, кратных 5= " << sm << endl;

}

}

****