Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №11.1**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Информационные и динамические структуры. Одно- и Двунаправленные списки

Вариант 24

Выполнила:

Студент группы ИВТ-20-2б

Ананина Арина Юрьевна

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

1. Постановка задачи

Сформировать однонаправленный и двунаправленный списки. Распечатать полученную структуру. Выполнить обработку структуры:

* Для однонаправленного списка- добавить в список элементы с номерами 1, 3, 5 и т. д.
* Для двунаправленного списка- удалить из списка последний элемент с четным информационным полем

Распечатать полученный результат. Удалить соответствующую структуру из памяти.

1. Анализ задачи

Однонаправленные списки.

1. Используются типы данных int, char\*.

2. Переменные типа int используются в качестве подсчета элементов в списке или счетчика для цикла. Тип данных char\* используется в структуре.

3. Данные будут представлены в виде списка, список будет организован с помощью структуры.

4. Поля, используемые в структуре:

struct List

{

char\* data; // cтрока

List\* next; // указатель на следующий элемент

};

5. Ввод элементов в список происходит с клавиатуры. Вывод списка организован с помощью функции , внутри функции цикл c условием окончания- ссылка на NULL.

6. С помощью функции add в список добавляются элементы с номерами 1, 3, 5 и т.д. Для этого в функции организован цикл с условием окончания – ссылка на NULL, внутри цикла с помощью оператора условия проверяется нечетность номера элемента и если условие истина, то записываются новые данные.

while (p != NULL)

{

if (p->data != NULL && i % 2 == 1) {

cout << "Введите " << i << " элемент: ";

cin >> p->data;

}

p = p->next;

i++;

Двунаправленные списки.

1. Используются типы данных int, char\*.

2. Переменные типа int используются в качестве подсчета элементов в списке или счетчика для цикла. Тип данных char\* используется в структуре.

3. Данные будут представлены в виде списка, список будет организован с помощью структуры.

4. Поля, используемые в структуре:

struct List

{

char\* data; // cтрока

List\* next; // указатель на следующий элемент

};

5. Ввод элементов в список происходит с клавиатуры. Вывод списка организован с помощью функции , внутри функции цикл c условием окончания- ссылка на NULL.

6. С помощью функции add в список добавляются элементы с номерами 1, 3, 5 и т.д. Для этого в функции организован цикл с условием окончания – ссылка на NULL, внутри цикла с помощью оператора условия проверяется нечетность номера элемента и если условие истина, то записываются новые данные.

while (p != NULL)

{

if (p->data != NULL && i % 2 == 1) {

cout << "Введите " << i << " элемент: ";

cin >> p->data;

}

p = p->next;

i++;

1. Код программы

Однонаправленные списки

#include <iostream>

#include <windows.h>

#include <string>

using namespace std;

struct List

{

char\* data;

List\* next;

};

List\* make(int n)

{

if (n == 0) {

cout << "Список не может быть создан" << endl;

return NULL;

}

List\* first, \* p;

first = NULL;

p = new List;

p->data = new char[];

cout << "Введите первый элемент: ";

cin >> p->data;

first = p;

for (int i = 2; i <= n; i++)

{

p->next = new List;

p = p->next;

p->data = new char[];

p->next = NULL;

cout << "Введите " << i << " элемент: ";

cin >> p->data;

}

return first;

}

void print(List\* first)

{

if (first == NULL)

{

cout << "Список пустой" << endl;

}

else

{

List\* p = first;

while (p != NULL)

{

if (p->data != NULL) {

cout << p->data << " ";

}

p = p->next;

}

cout << endl;

}

}

void add(List\* list)

{

if (list == NULL)

{

cout << "Список пустой" << endl;

}

else

{

int i = 1;

List\* p = list;

while (p != NULL)

{

if (p->data != NULL && i % 2 == 1) {

cout << "Введите " << i << "элемент";

cin >> p->data;

}

p = p->next;

i++;

}

cout << endl;

}

}

int main()

{

setlocale(LC\_ALL, "RUS");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int size;

do

{

cout << "Введите количество элементов списка" << endl;

cin >> size;

} while (size < 1);

List\* list = make(size);

print(list);

add(list);

print(list);

return 0;

}

Двунаправленные списки

#include <iostream>

#include <string>

#include <Windows.h>

#include <iterator>

#include <list>

using namespace std;

// с помощью STL библиотеки

list<int> makeSTL(int n)

{

list<int> res;

int tmp;

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

{

cout << "Введите элемент " << i + 1 << ": ";

cin >> tmp;

res.push\_back(tmp);

}

return res;

}

void printSTL(list<int> a)

{

if (a.size() == 0)

cout << "Cписок пустой" << endl;

else

{

for (list<int>::iterator it = a.begin(); it != a.end(); it++)

cout << \*it << " ";

cout << endl;

}

}

void deletionSTL(list<int> &a)

{

if (a.size() == 0)

cout << "Cписок пустой" << endl;

else

{

bool f = true;

list<int>::reverse\_iterator itr = a.rbegin();

while (itr != a.rend())

{

if (\*itr % 2 == 0)

{

a.erase(next(itr).base());

break;

}

itr++;

}

cout << endl;

}

}

// с помощью структур

struct List

{

int data;

List\* next;

List\* prev;

};

List\* make(int n)

{

if (n == 0) {

cout << "Список не может быть создан" << endl;

return NULL;

}

List\* first, \* p;

first = NULL;

cout << "Введите 1 элемент: ";

p = new List;

cin >> p->data;

first = p;

p->prev = NULL;

for (int i = 2; i <= n; i++)

{

p->next = new List;

p->next->prev = p;

p = p->next;

cout << "Введите " << i << " элемент: ";

cin >> p->data;

p->next = NULL;

}

return first;

}

void print(List\* first)

{

if (first == NULL)

{

cout << "Список пустой" << endl;

}

else

{

List\* p = first;

while (p != NULL)

{

if (p->data != NULL) {

cout << p->data << " ";

}

p = p->next;

}

cout << endl;

}

}

void deletion(List\* list)

{

if (list == NULL)

{

cout << "Список пустой" << endl;

}

else

{

List\* evenItem = NULL;

while (list != NULL)

{

if (list->data != NULL && list->data % 2 == 0) {

evenItem = list;

}

list = list->next;

}

if (evenItem != NULL) {

if (evenItem->prev != NULL) {

evenItem->prev->next = evenItem->next;

}

if (evenItem->next != NULL) {

evenItem->next->prev = evenItem->prev;

}

}

cout << endl;

}

}

int main()

{

system("color F0");

setlocale(LC\_ALL, "RUS");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

// с помощью STL

cout << "Двусвязный список через STL" << endl;

int size;

do

{

cout << "Введите количество элементов списка ";

cin >> size;

} while (size < 1);

list<int> myList = makeSTL(size);

printSTL(myList);

deletionSTL(myList);

printSTL(myList);

// с помощью структур

int size2;

do

{

cout << "Введите количество элементов списка" << endl;

cin >> size2;

} while (size2 < 1);

List\* list = make(size2);

List\* list2;

print(list);

deletion(list);

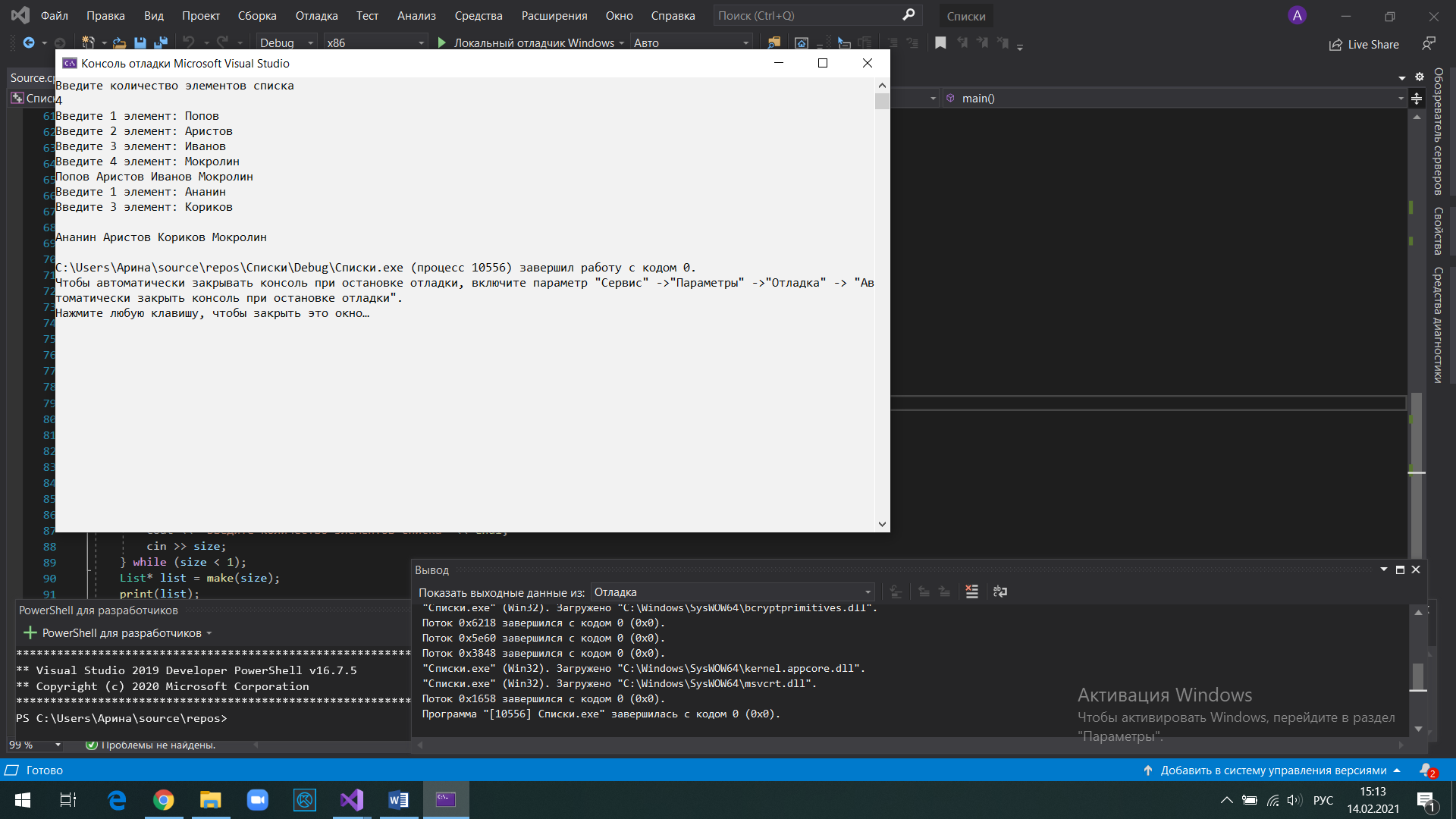
print(list);

return 0;

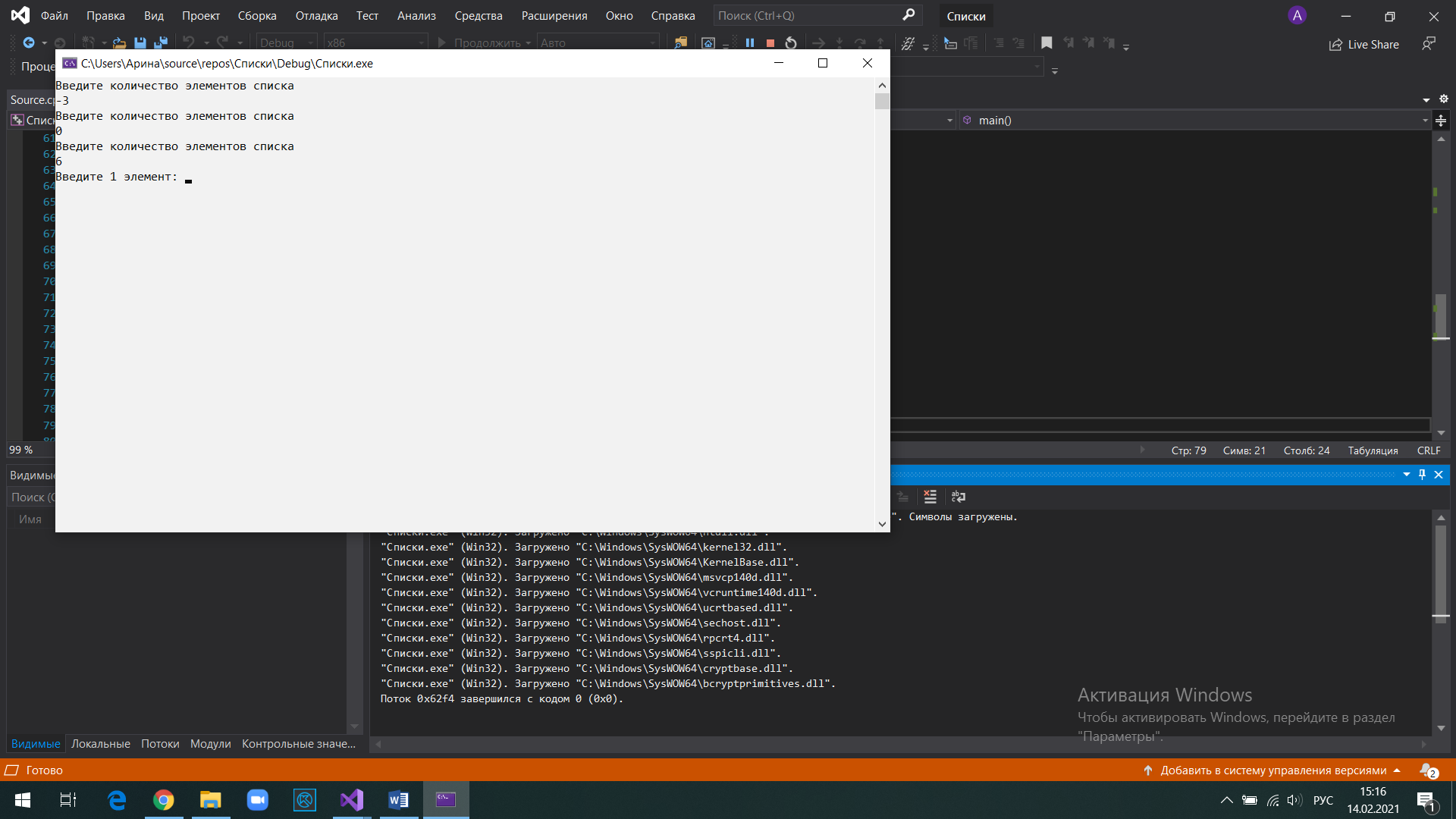
}

4. Выполнение программы.

Однонаправленные списки:



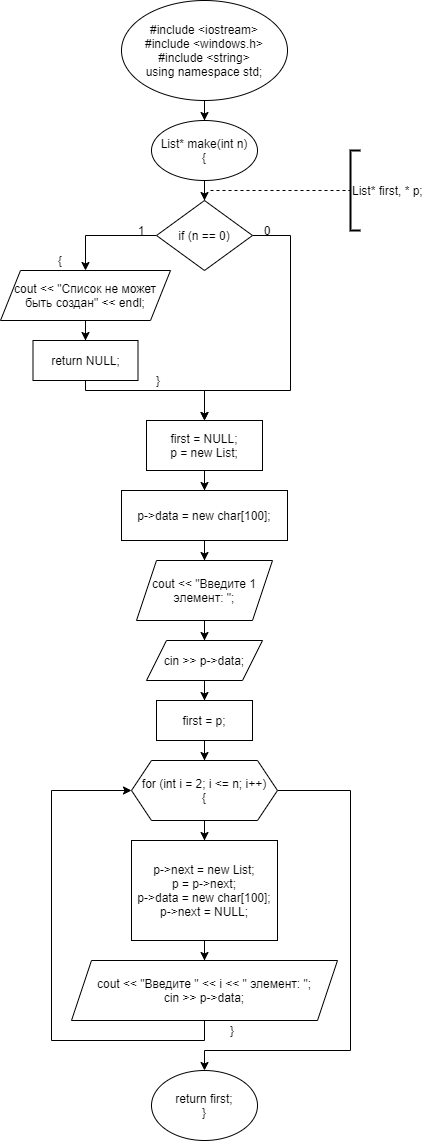
Если в качестве кол-ва элементов пользователь вводит отрицательное число, то программа будет запрашивать ввести его снова до тех пор, пока не будет введено корректное значение.

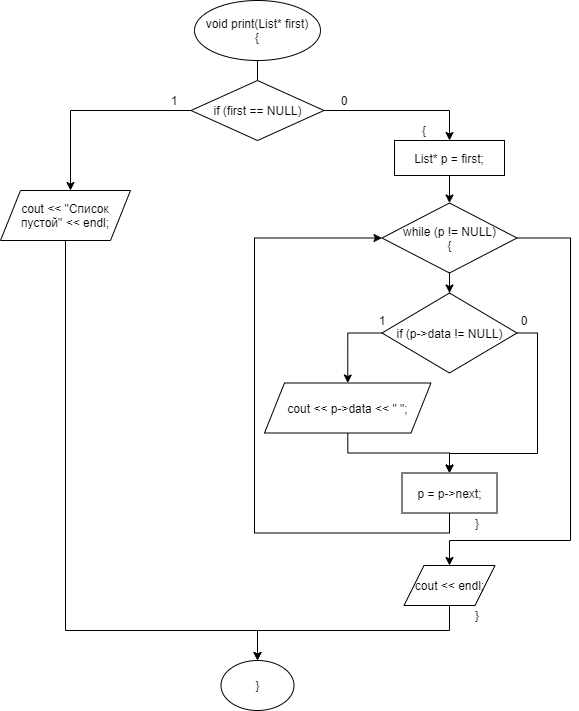


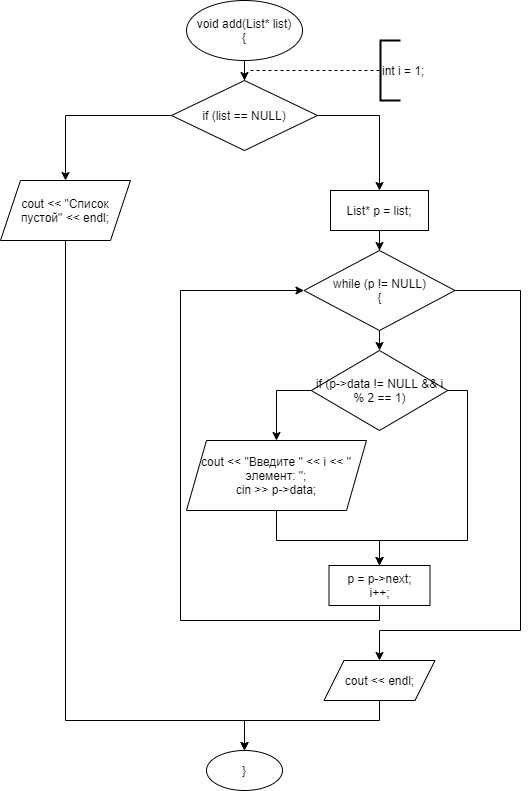
Двунаправленные списки:

5. Блок схема

Однонаправленные списки







Двунаправленные списки

