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 4

Выполнил:

Студент группы РИС-20-1б

Еске Вячеслав Сергеевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Получить практичeские навыки работы с однонаправленными списками.

**Постановка задачи**

1. Однонаправленный список - Тип информационного поля int. Удалить из списка последний элемент с четным информационным полем.  
Двунаправленный список - Тип информационного поля char\*. Добавить в список элемент с заданным номером.

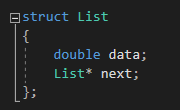
2. Распечатать полученную структуру.

3. Выполнить обработку структуры в соответствии с заданием.

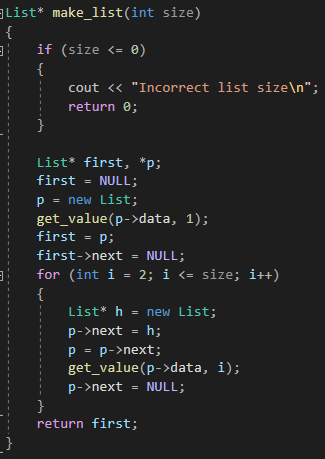
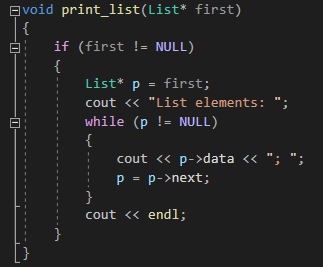
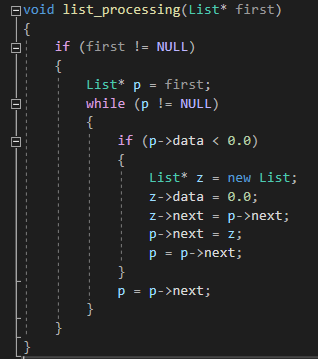
4. Распечатать полученный результат.

5. Удалить соответствующую структуру из памяти.

**Анализ задачи**

1. Определить какие действия необходимо выполнить для решения задачи.
   1. Реализация списка производится через структуру List. 
   2. Создать однонаправленный список.



* 1. Заполнить список вещественными числами. 
  2. Вывести все элементы списка в консоль. 
  3. Далее необходимо найти элементы с отрицательными значениями, и перед ними вставить элемент со значением равным 0. 
  4. В конце выводятся все элементы списка.

1. Для решения задачи использовались следующие типы данных.

2.1 Данные элементов списка хранятся в формате double.



2.2 Указатель на следующий элемент является экземпляром структуры. 

**Код**

#include <iostream>

using namespace std;

struct List {

int data;

List\* next;

};

List\* make(int size) {

List\* first, \* p;

first = NULL;

cout << "Введите 1 элемент ";

p = new List;

cin >> p->data;

first = p;

for (int i = 2; i <= size; i++) {

List\* h = new List;

p->next = h;

p = p->next;

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

cin >> p->data;

p->next = NULL;

}

return first;

}

void print(List\* first) {

if (first == NULL) {

cout << "Список пуст" << endl;

}

else {

List\* p = first;

do {

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

p = p->next;

} while (p != NULL);

cout << endl;

}

}

void del(List\*& first) {

List\* p = first;

List\* q = new List;

while (p != NULL) {

if (p->data % 2 == 0) q = p;

p = p->next;

}

p = first;

if (p == q) {

p = p->next;

delete first;

first = p;

}

else {

while (p->next != NULL) {

if (p->next == q) {

List\* x = p->next;

p->next = p->next->next;

delete x;

}

else p = p->next;

}

}

p = first;

}

int main() {

setlocale(LC\_ALL, "RUS");

int size;

do {

cout << "Размер списка: ";

cin >> size;

} while (size <= 0);

List\* list = make(size);

cout << "Ваш список " << endl;

print(list);

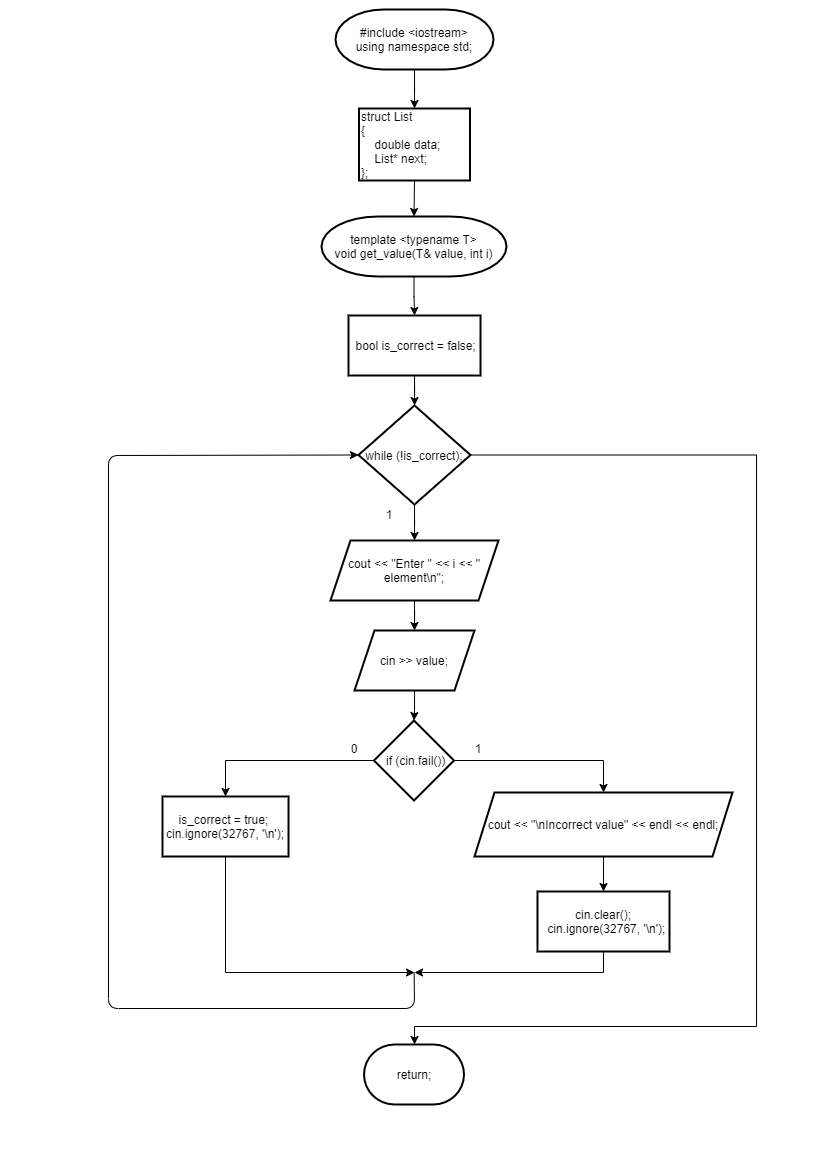
del(list);

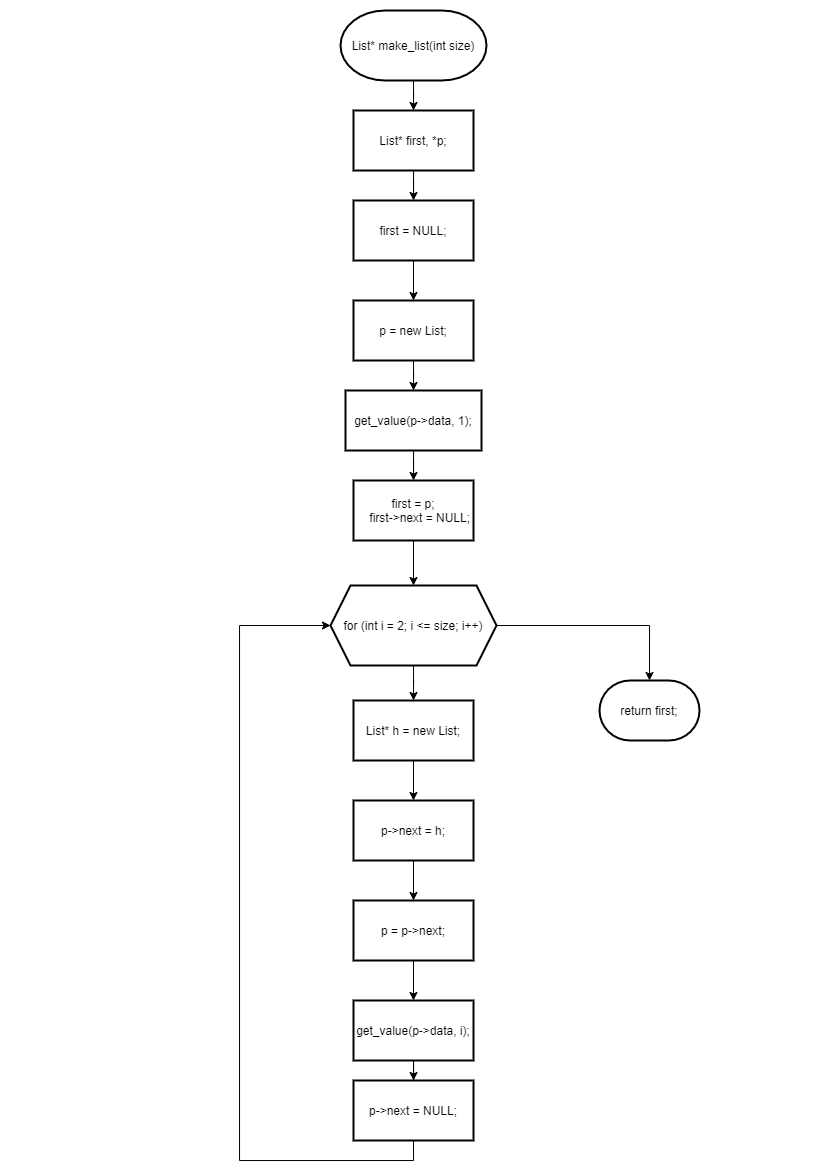
print(list);

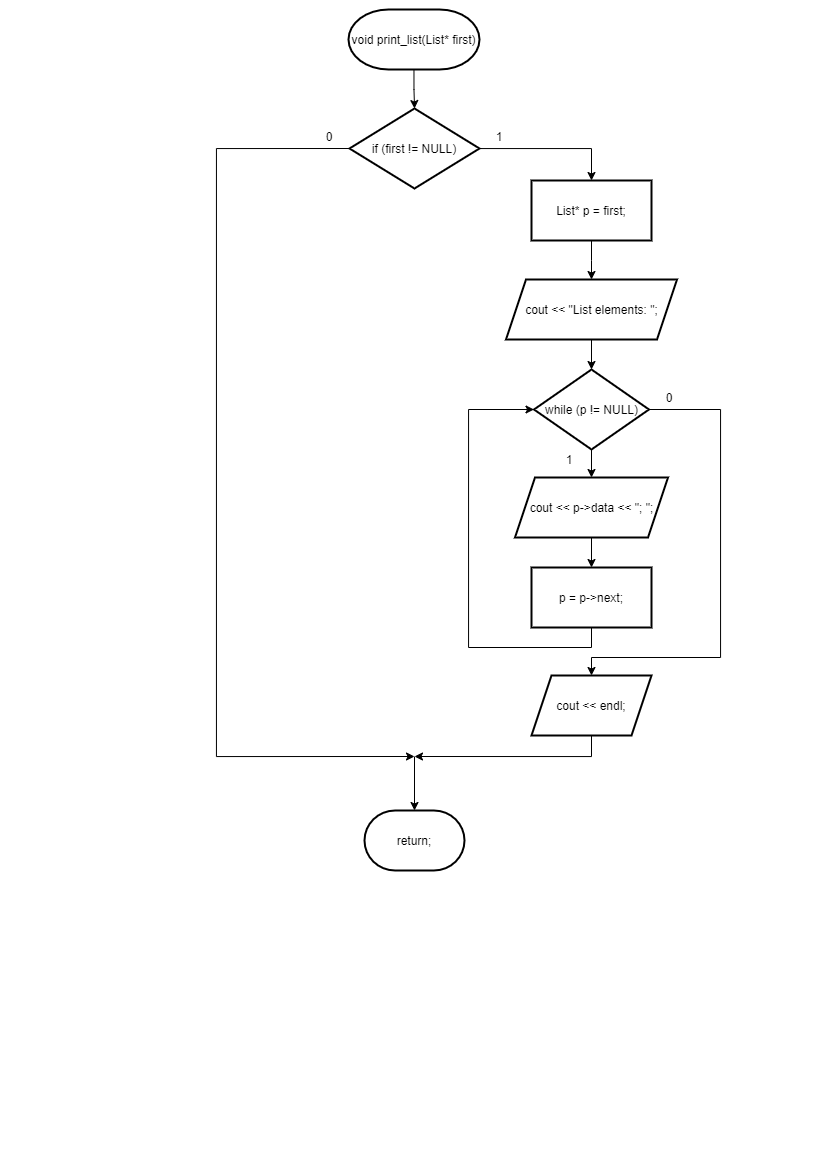
return 0;

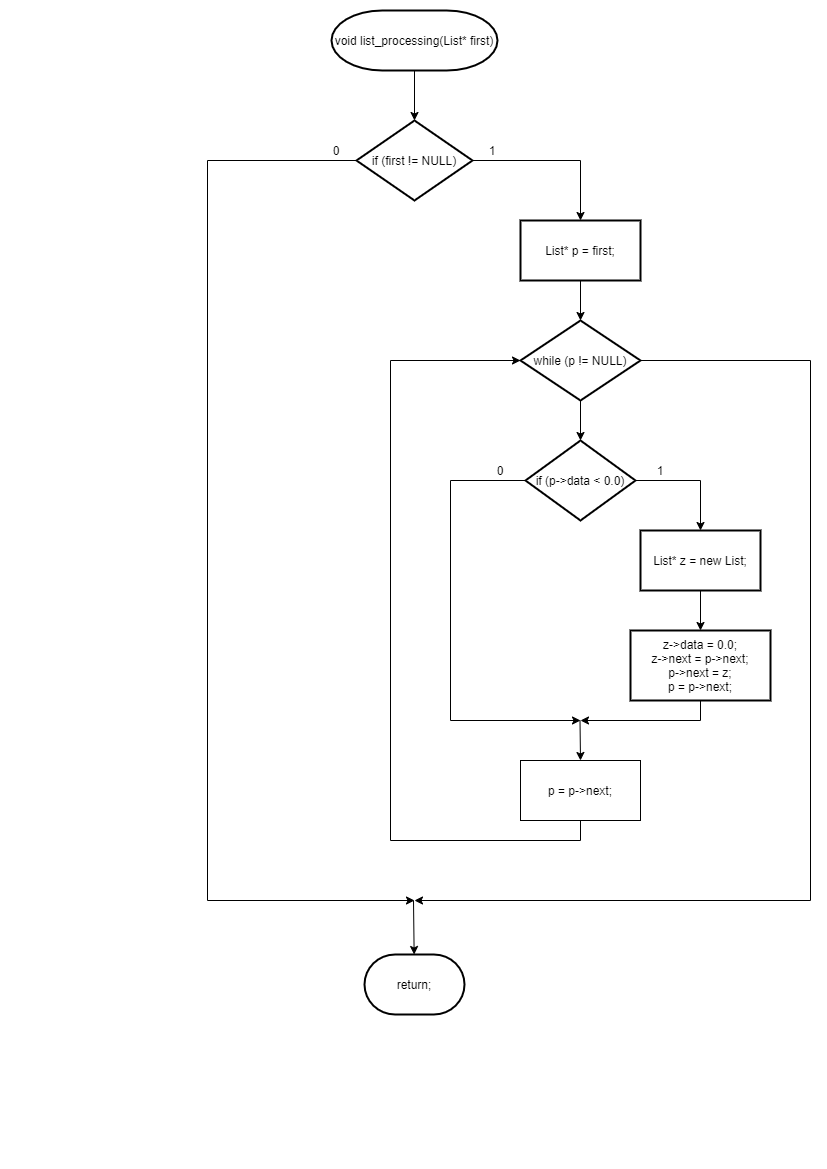
}

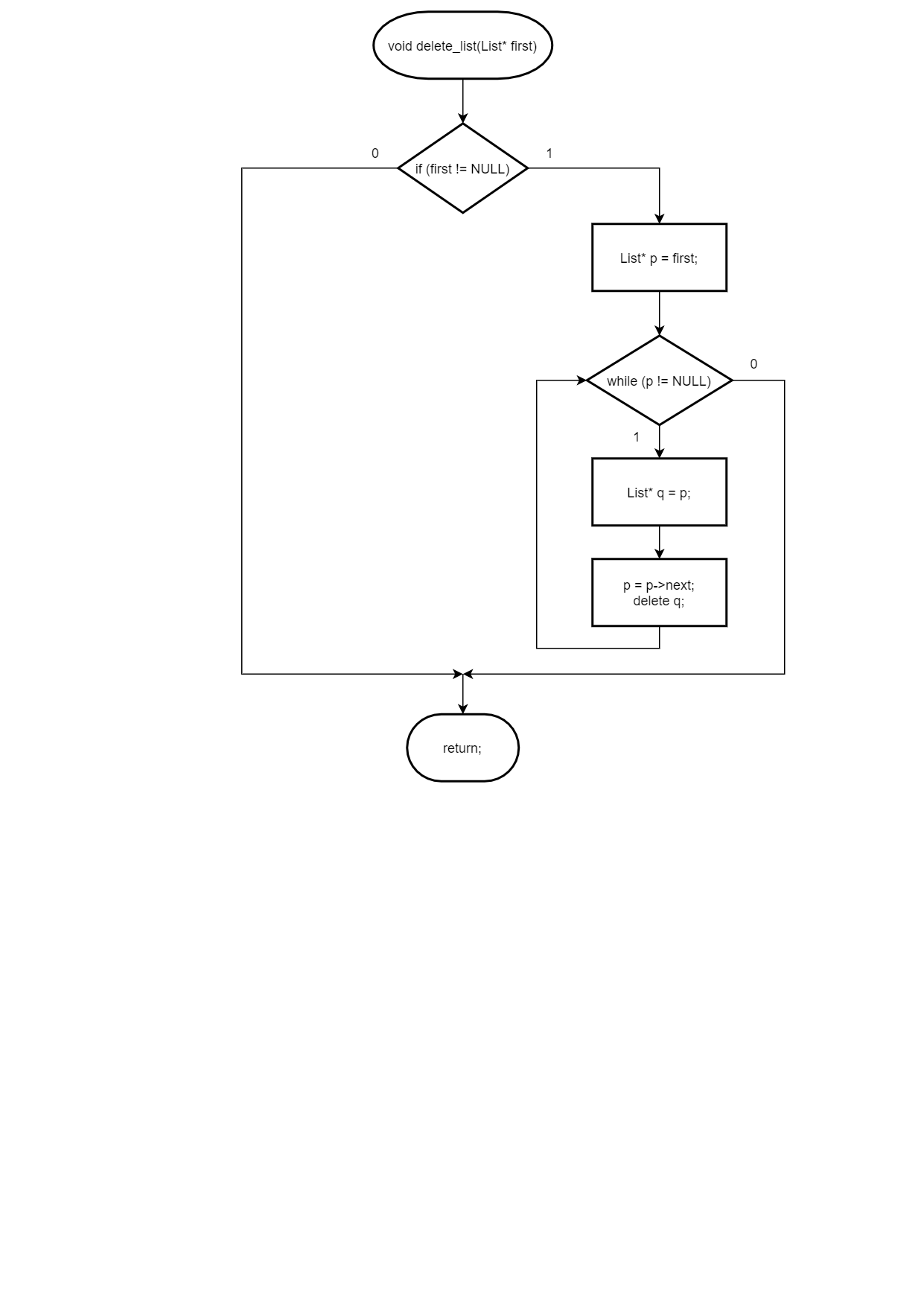
**Блок-схема**

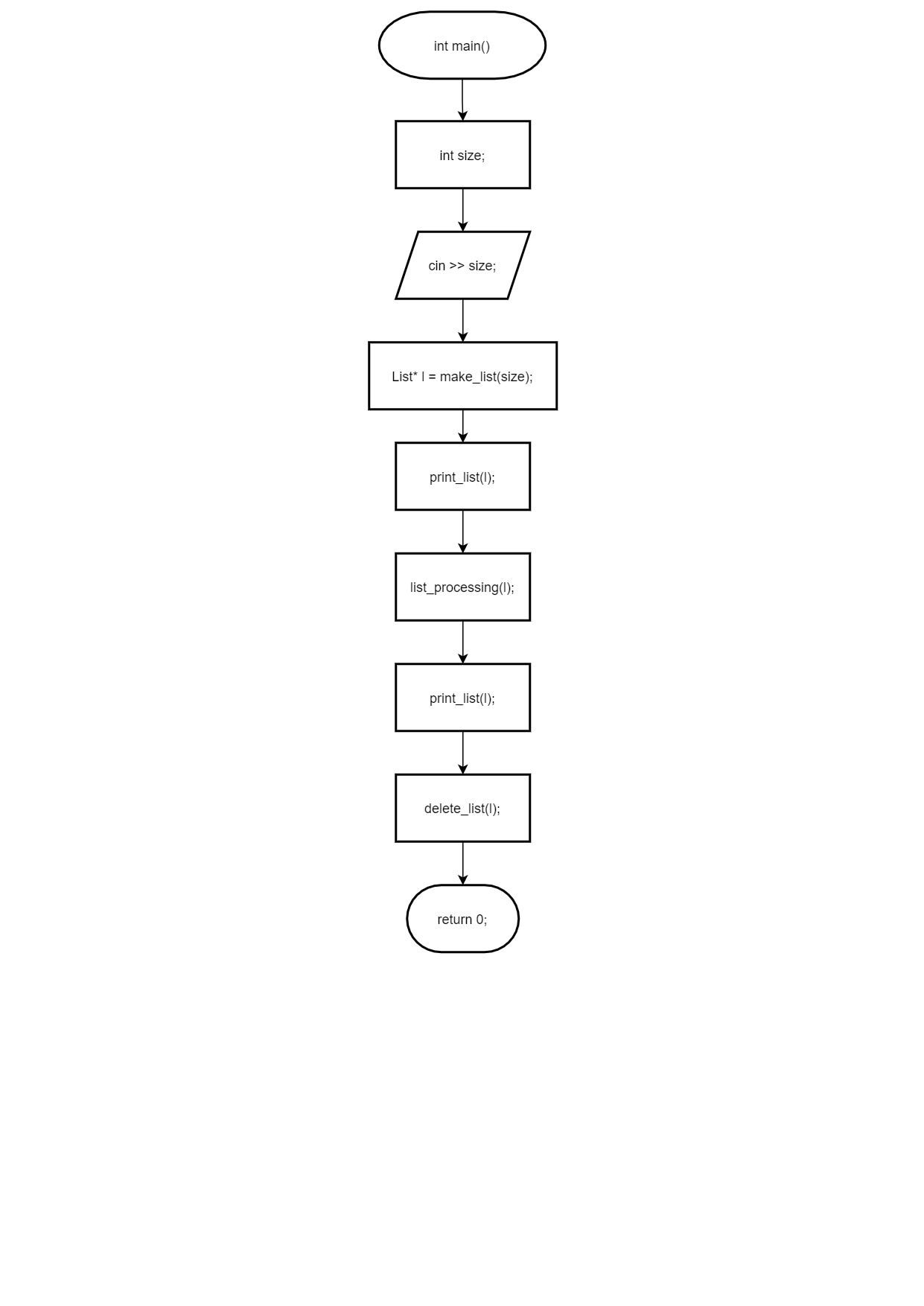
****

****

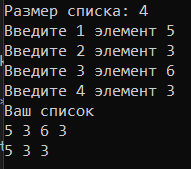
****

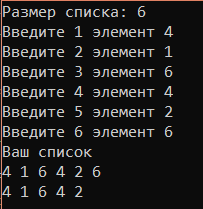
****

****

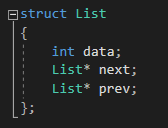
****

**Скриншоты тестов №1**

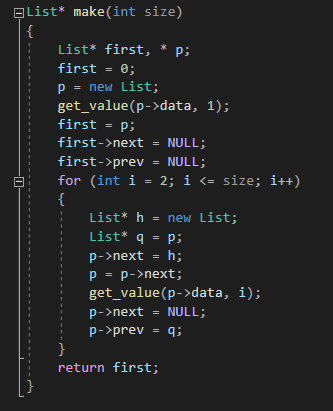
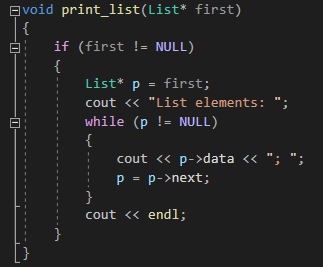
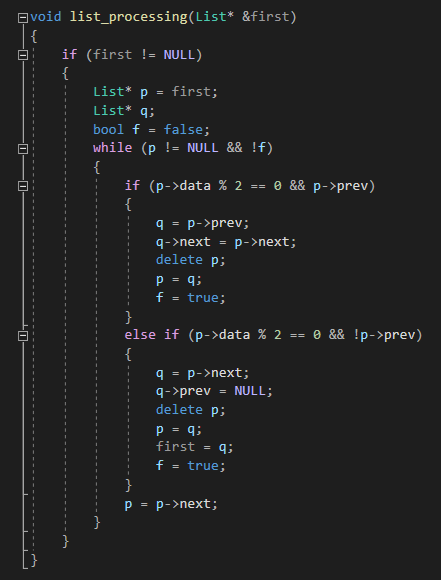
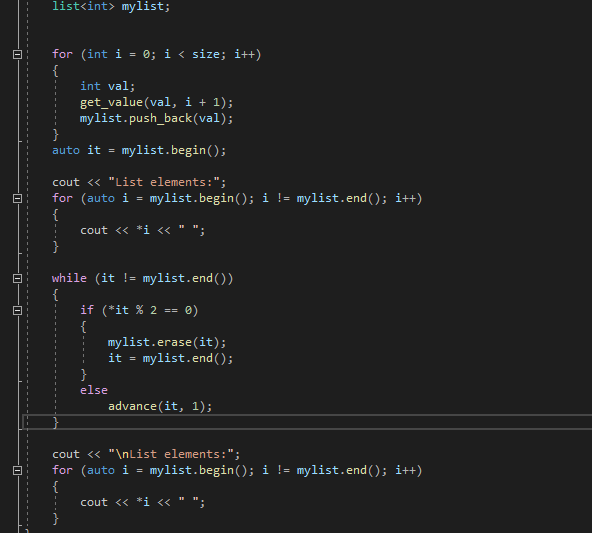




**Анализ задачи №2**

1. Определить какие действия необходимо выполнить для решения задачи.
   1. Реализация списка производится через структуру List. 
   2. Создать двунаправленный список.



* 1. Заполнить список целыми числами. 
  2. Вывести все элементы списка в консоль. 
  3. Далее необходимо найти первый чётный элемент. 
  4. В конце выводятся все элементы списка.
  5. Реализация через STL. 

1. Для решения задачи использовались следующие типы данных.

2.1 Данные элементов списка хранятся в формате int.



2.2 Указатель на следующий и предыдущий элемент является экземпляром структуры.



**Код №2**

#include <iostream>

using namespace std;

struct List {

char data;

List\* next;

List\* prev;

};

List\* make(int size) {

List\* first, \* p;

first = NULL;

p = new List;

cout << "Введите 1 элемент ";

p = new List;

cin >> p->data;

first = p;

p->next = NULL;

p->prev = NULL;

for (int i = 2; i <= size; i++) {

List\* h = new List;

List\* q = p;

p->next = h;

p = p->next;

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

cin >> p->data;

p->next = NULL;

p->prev = q;

}

return first;

}

void print(List\* first) {

if (first == NULL) {

cout << "Список пуст" << endl;

}

else {

List\* p = first;

while (p != NULL) {

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

p = p->next;

}

cout << endl;

}

}

void add(List\*& top, int pos, int x)

{

int j = 1;

int i; List\* pnew, \* p = top;

pnew = new List; pnew->data = x;

if (pos <= 1 || top == NULL)

{

pnew->next = top; top = pnew;

}

else

{

for (i = 1; i < pos - 1; i++)

if (p->next != NULL)

{

p = p->next;

j++;

}

if (pos <= j + 1)

{

pnew->next = p->next; p->next = pnew;

}

}

}

int main() {

setlocale(LC\_ALL, "Rus");

int size,pos;

char x;

do {

cout << "Размер списка: ";

cin >> size;

} while (size <= 0);

List\* list = make(size);

cout << "Ваш список " << endl;

print(list);

do {

cout << "Введите номер позиции для вставки: ";

cin >> pos;

} while ((pos < 1) || (pos > size + 1));

cout << "Введиде значение элемента для вставки: ";

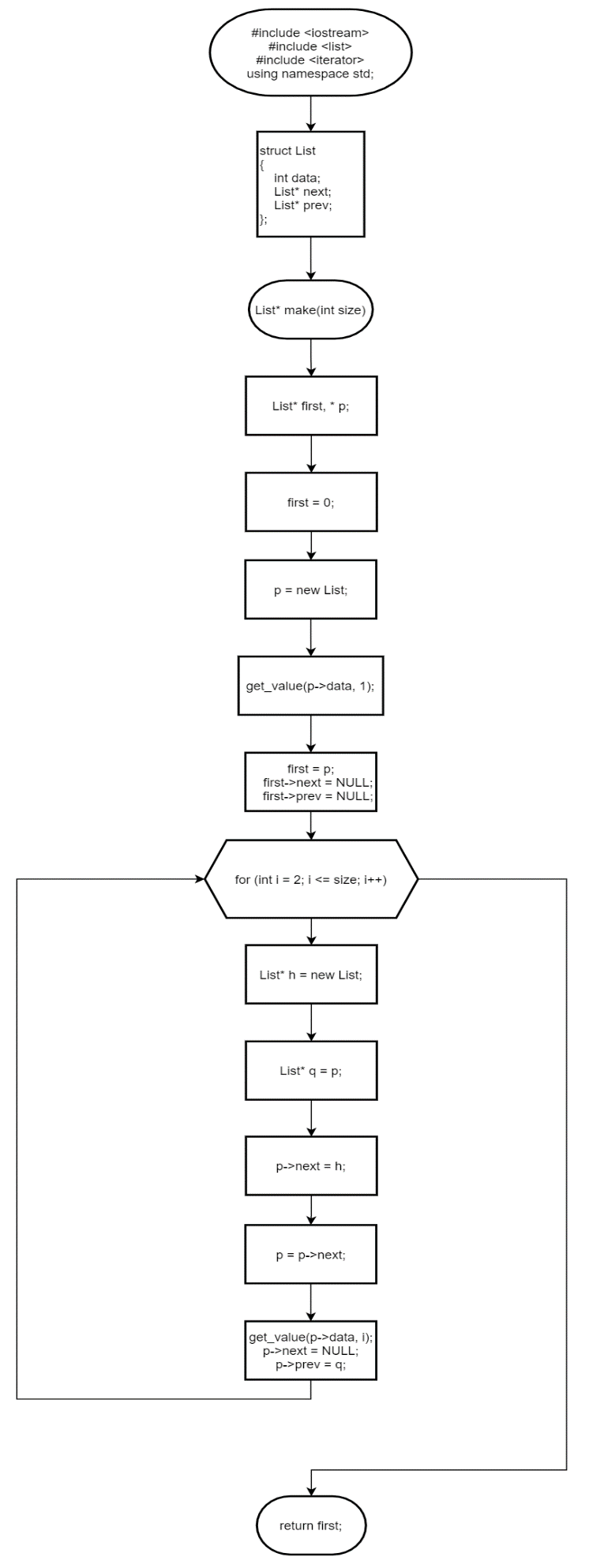
cin >> x;

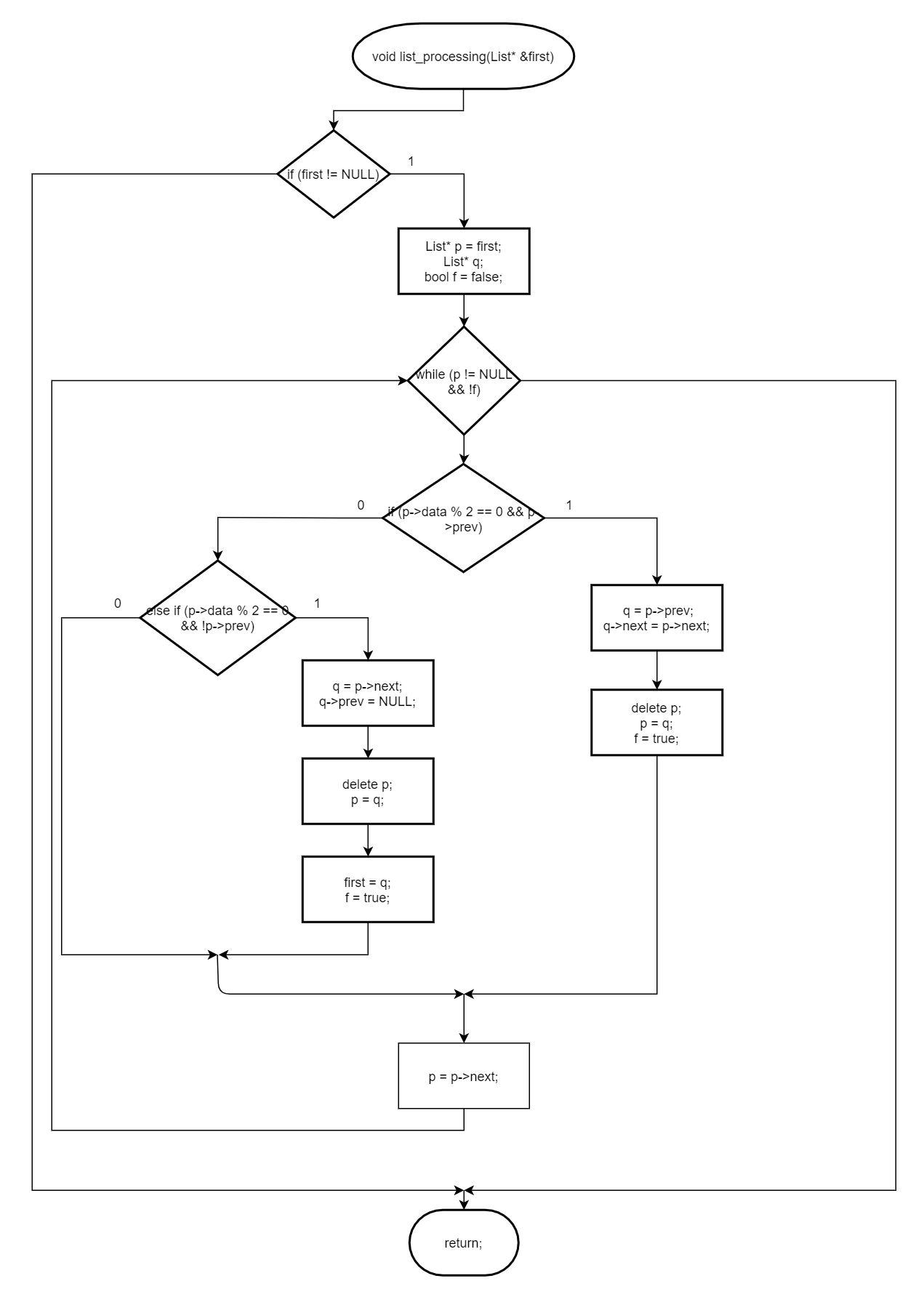
add(list, pos, x);

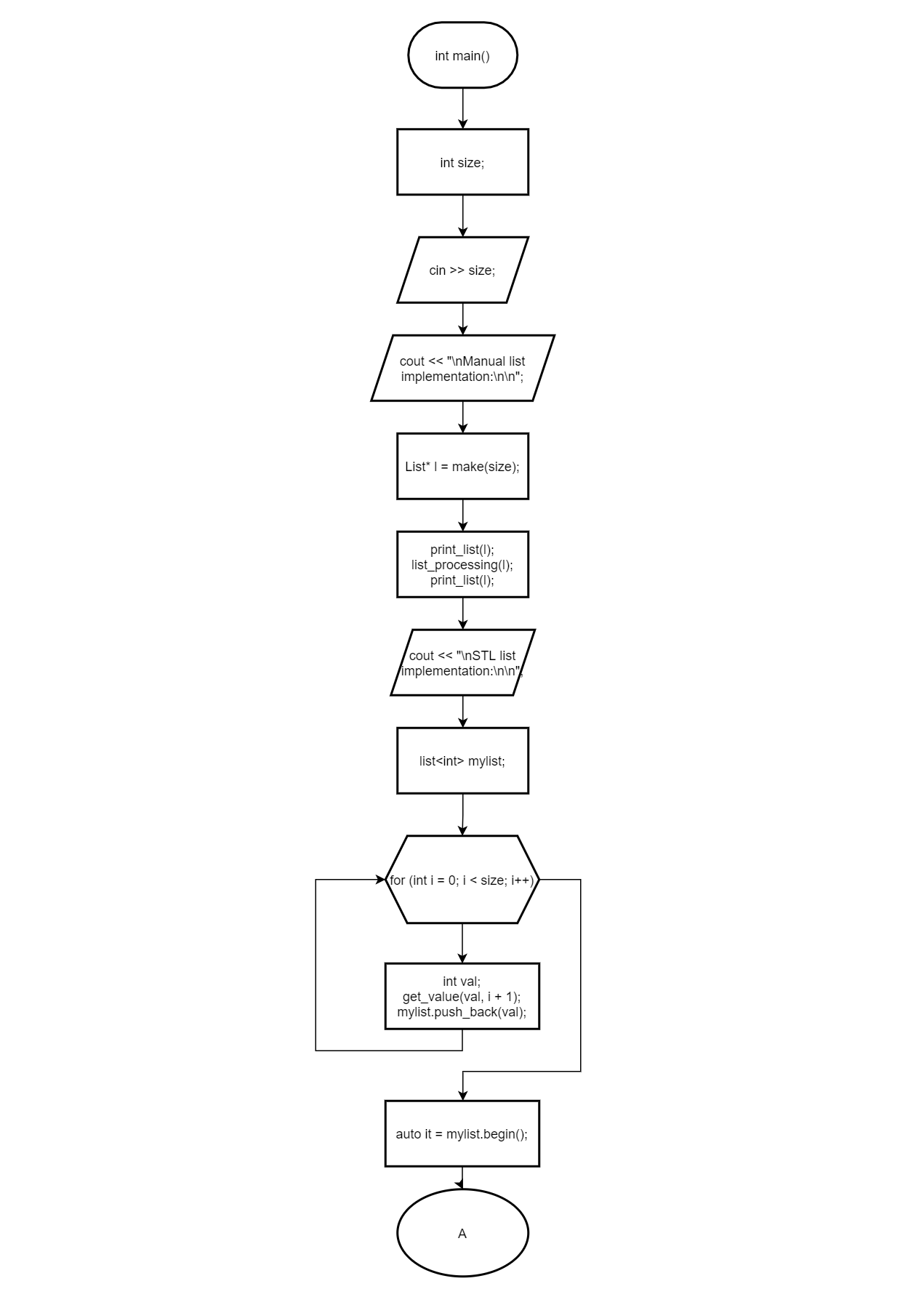
print(list);

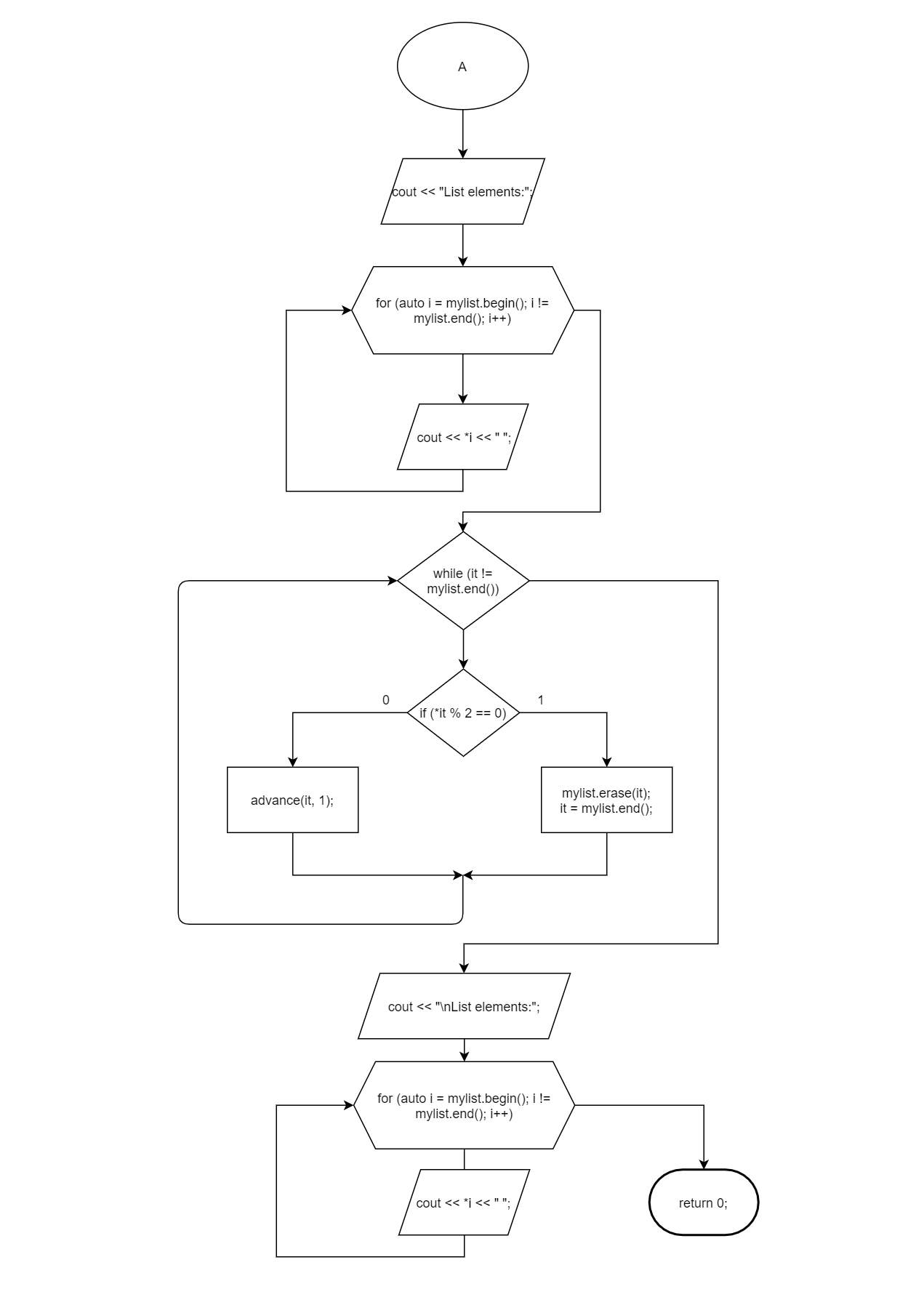
}

**Блок схема №2**

****

****

****

****

**Тесты №2**

