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 №11.2**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: Информационные и динамические структуры. Очереди.

Вариант 4

Выполнил:

Студент группы РИС-20-1б

Еске Вячеслав Сергеевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Получить практические навыки работы с очередями.

**Постановка задачи**

1. Тип информационного поля char\*. Добавить в очередь элемент с заданным номером.

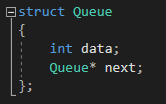
2. Распечатать полученную структуру.

3. Выполнить обработку структуры в соответствии с заданием.

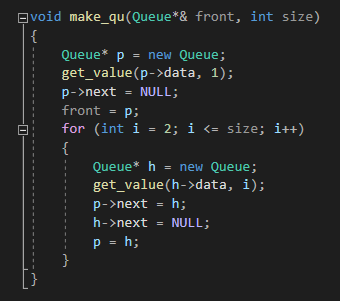
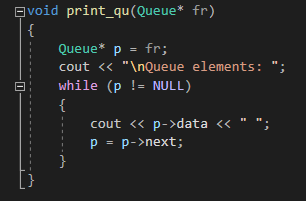
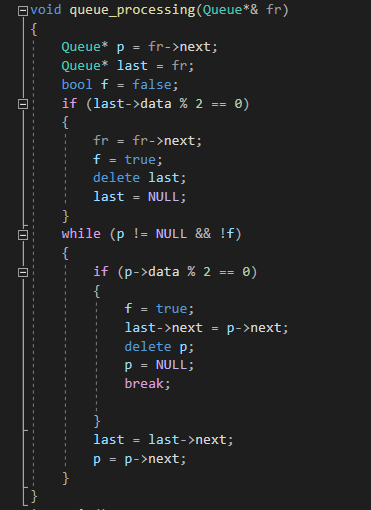
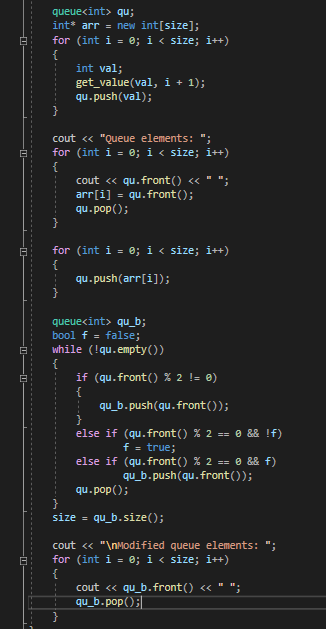
4. Распечатать полученный результат.

5. Удалить соответствующую структуру из памяти.

**Анализ задачи**

1. Определить какие действия необходимо выполнить для решения задачи.
   1. Реализация очереди производится через структуру Queue. 
   2. Создать очередь.



* 1. Заполнить очередь целыми числами. 
  2. Вывести все элементы очереди в консоль. 
  3. Поиск первого чётного элемента очереди. 
  4. 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 элемент является экземпляром структуры.



**Код**

#include <iostream>

using namespace std;

struct List

{

char data;

List\* next;

};

List\* make\_list(int n, List\*& first, List\*& last) {

if (n == 0) return NULL;

List\* p = new List;

char a;

cin >> a;

p->data = a;

p->next = NULL;

first = p;

last = p;

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

List\* h = new List;

cin >> a;

h->data = a;

h->next = p;

last = h;

p = last;

}

return first;

}

void addToPos(const int value, int Pos, List\*& last) //добавление в указанную

{ //позицию

List\* tmp = last;

List\* cur = NULL;

List\* Ptr = new List;

Ptr->data = value;

if (Pos == 1) {

Ptr->data = value;

Ptr->next = last;

last = Ptr;

}

else

{

int Kol\_el = 1;

while (tmp != NULL && Kol\_el != Pos)

{

cur = tmp;

tmp = tmp->next;

++Kol\_el;

}

cur->next = Ptr;

Ptr->next = tmp;

}

}

void print\_list(List\* last)

{

List\* p = last;

while (p != NULL) {

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

p = p->next;

}

}

int main() {

setlocale(LC\_ALL, "rus");

int n,l;

char a;

List\* first, \* last;

do {

cout << "Введите количество элементов: ";

cin >> n;

} while (n <= 0);

cout << "Введите элементы: ";

make\_list(n, first, last);

print\_list(last);

do {

cout << endl << "Введите номер добавляемого элемента: ";

cin >> l;

} while ((l < 1) || (l>n+1));

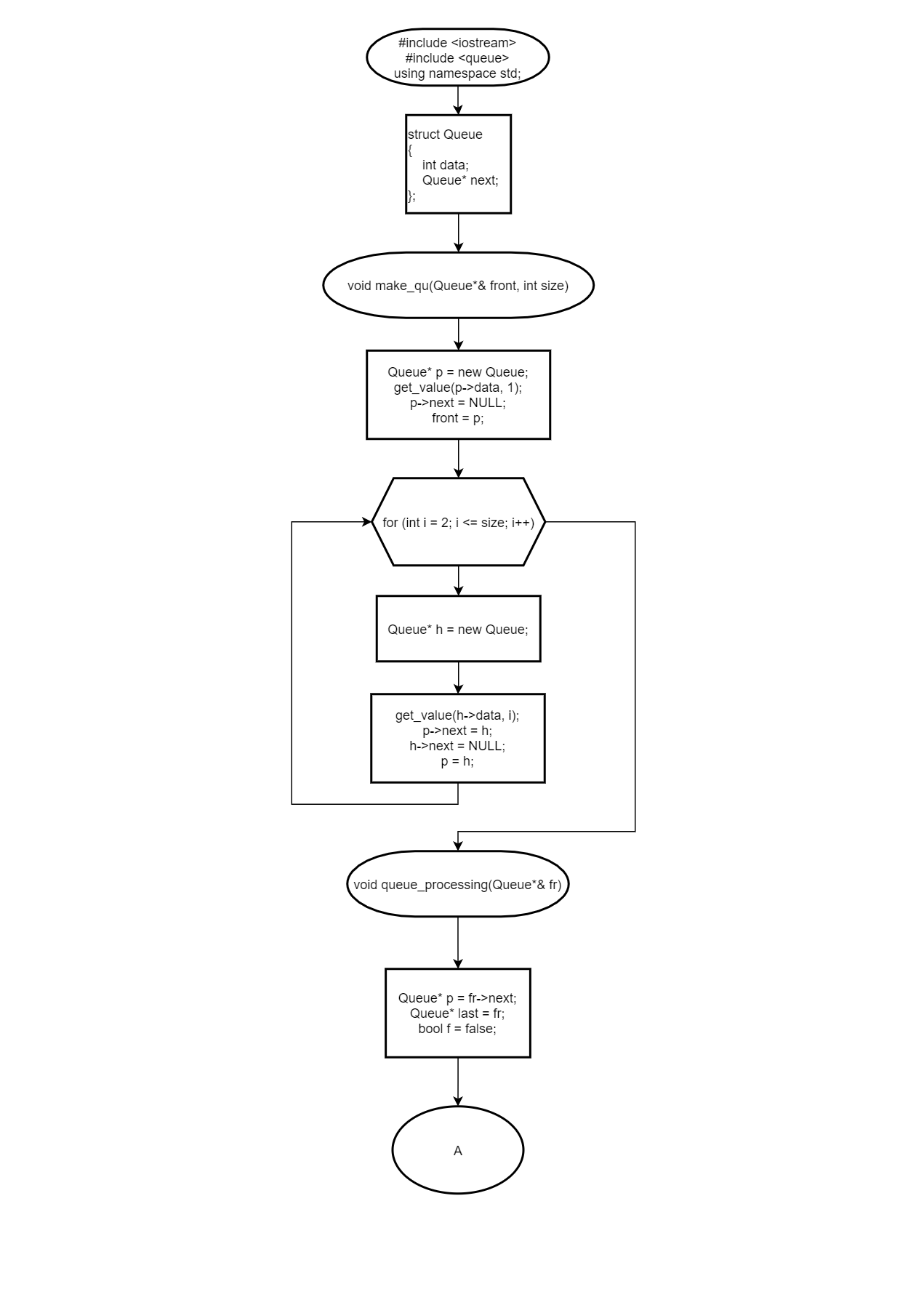
cout << "Введите добавляемый элемент: ";

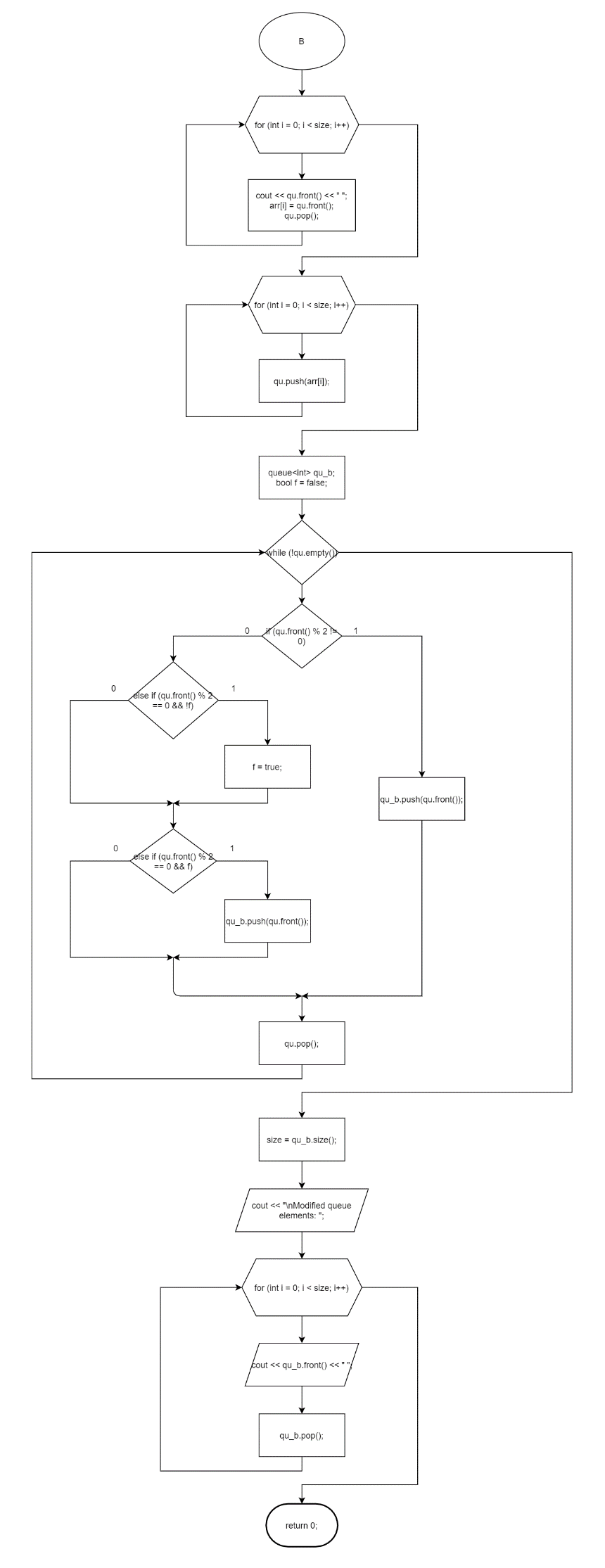
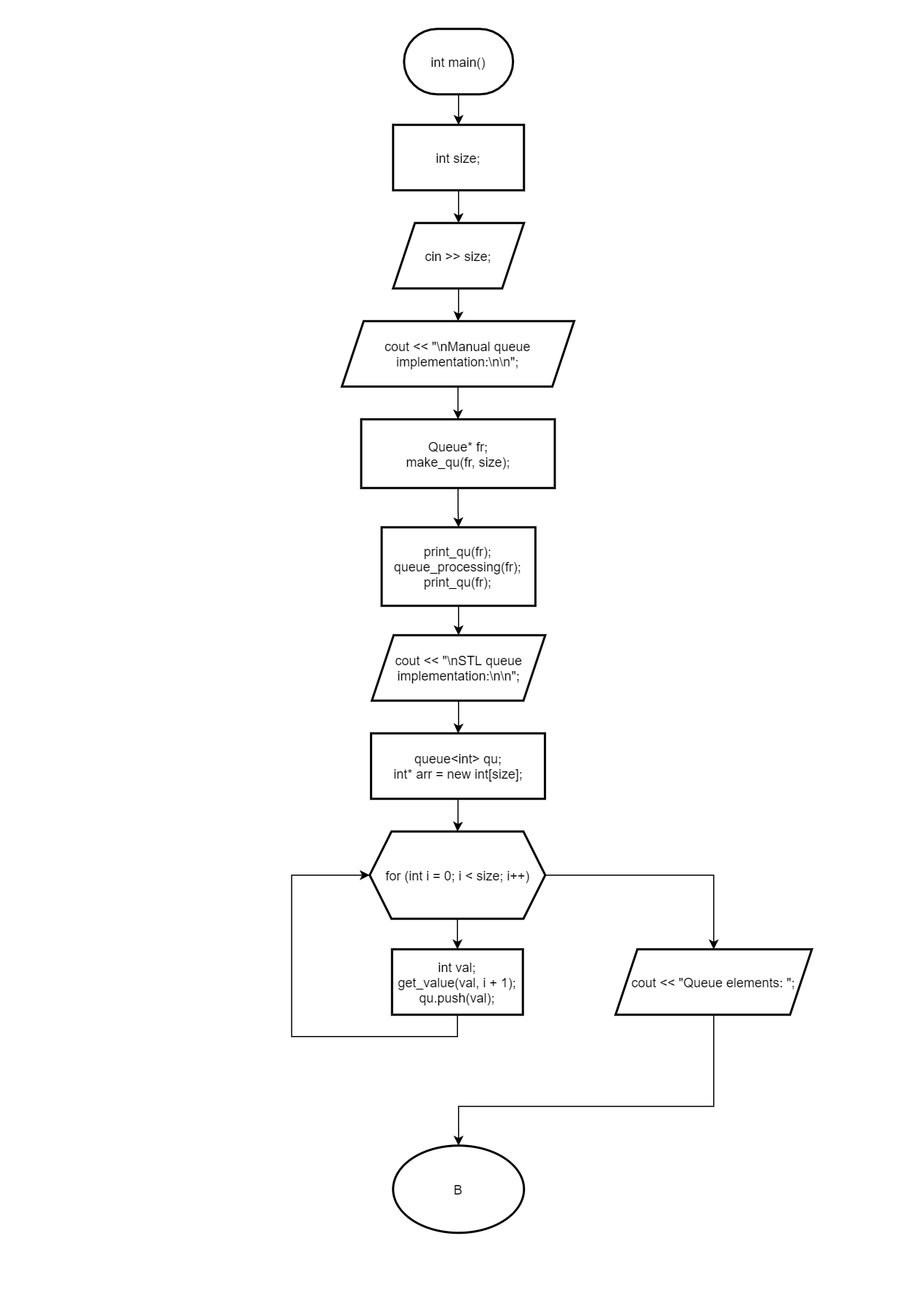
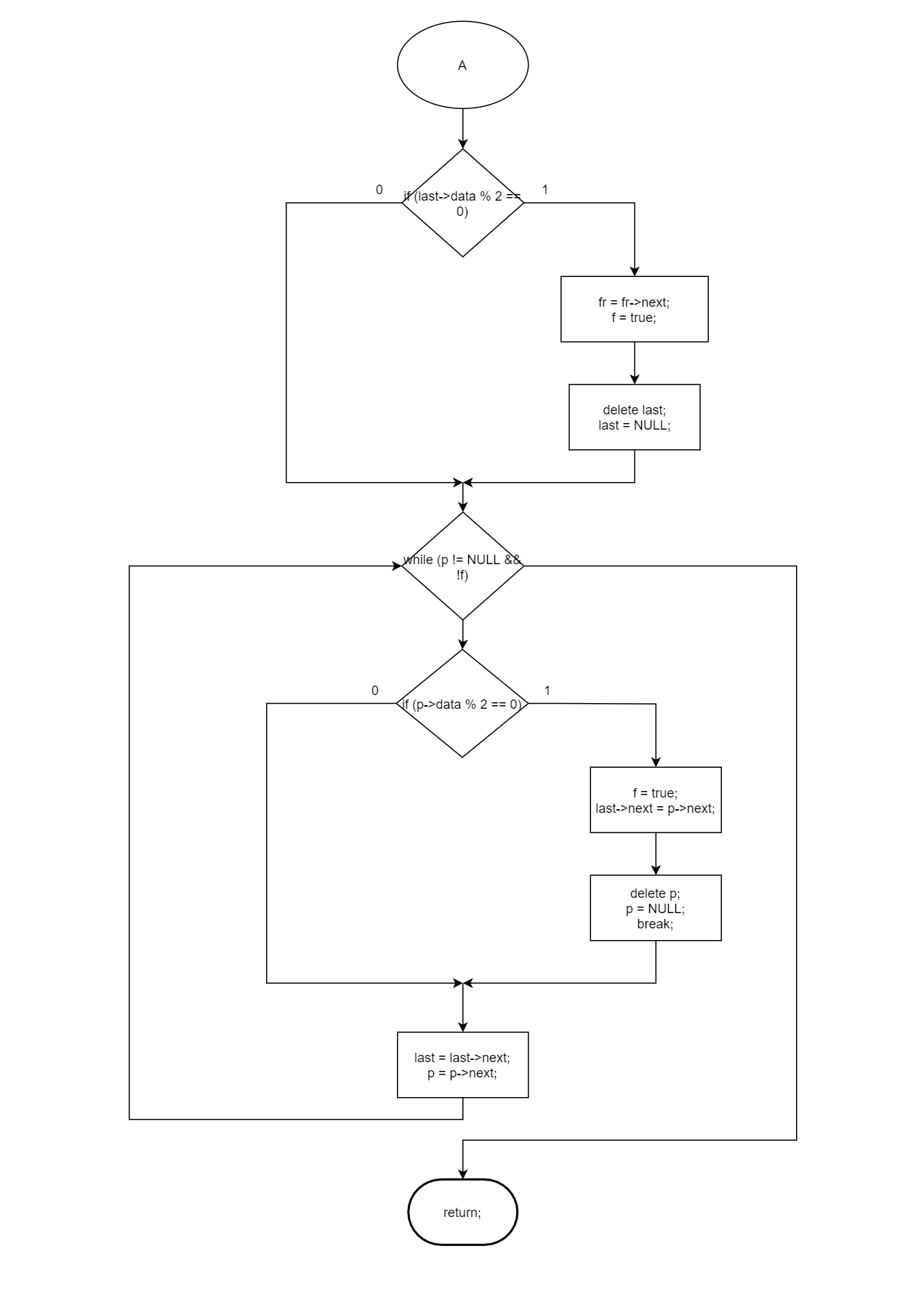
cin >> a;

addToPos(a, l, last);

print\_list(last);

}

**Блок схема **

****

**Тесты №3**

