Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №11.1**

Дисциплина: «основы алгоритмизации и программирования»

Тема: Информационные и динамические структуры. Одно— и двунаправленные списки

Выполнил работу

студент группы РИС-20-1б

Зверев А.Д.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

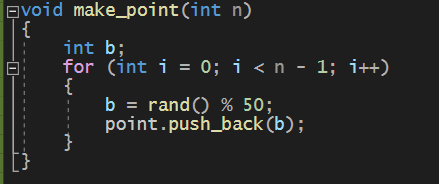
Пермь, 2021

**Постановка задачи**

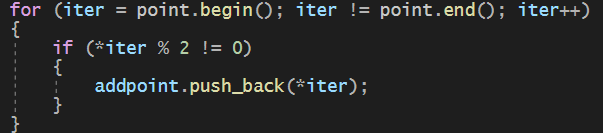
1. Сформировать двунаправленный список типа int.
2. Распечатать полученную структуру.
3. Удалить из списка все четные элементы.
4. Распечатать полученный результат.

**Анализ задачи**

1. Для решения задачи необходимо…
   1. Написать функцию make\_point типа void, которая будет заполнять список, размерность которого вводится пользователем;



* 1. Разработать и применить в основном теле программы цикл for, который будет удалять четные элементы из списка;

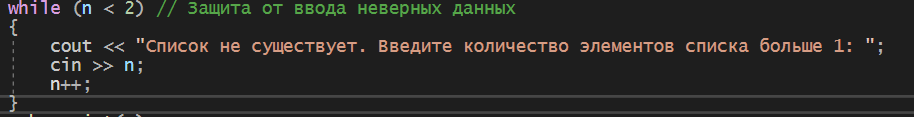


* 1. Вызвать все необходимые функции в основном теле программы;

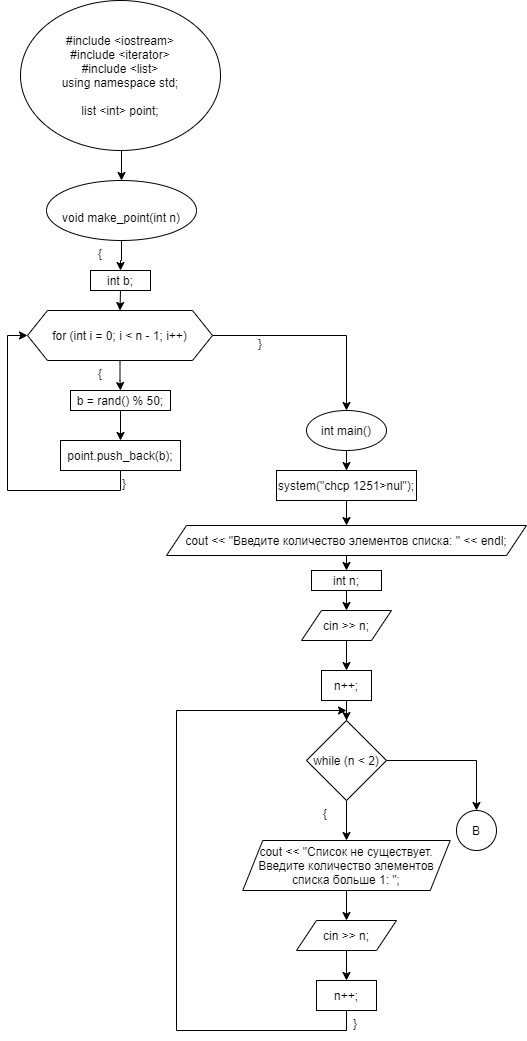


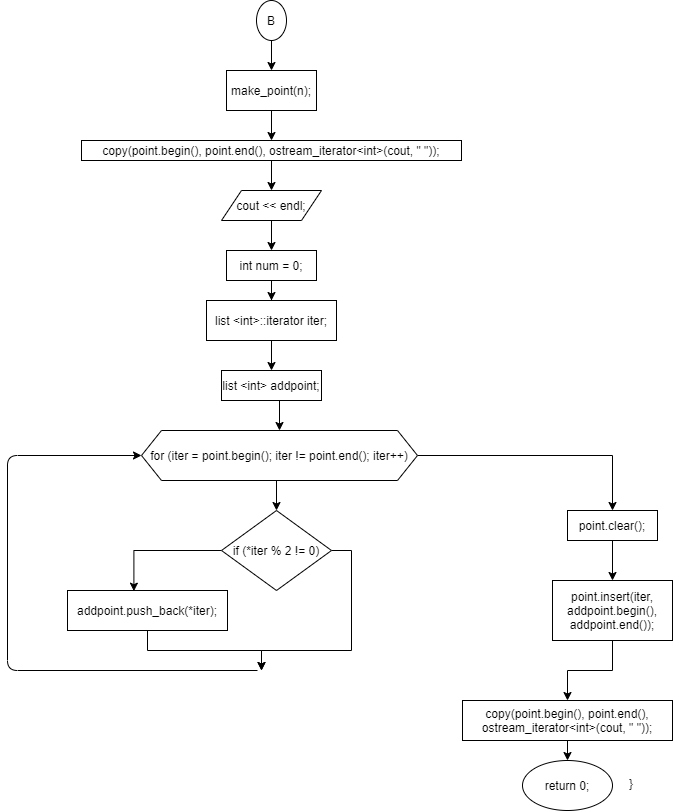
1. Для решения задачи были использованы данные следующих типов:
   1. Int для хранения количества элементов списка;





**Блок—схема программы**





**Решение**

#include <iostream>

#include <iterator>

#include <list>

using namespace std;

list <int> point;

void make\_point(int n)

{

int b;

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

{

b = rand() % 50;

point.push\_back(b);

}

}

int main()

{

system("chcp 1251>nul");

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

int n;

cin >> n;

n++;

while (n < 2) // Защита от ввода неверных данных

{

cout << "Список не существует. Введите количество элементов списка больше 1: ";

cin >> n;

}

make\_point(n);

copy(point.begin(), point.end(), ostream\_iterator<int>(cout, " "));

cout << endl;

int num = 0;

list <int>::iterator iter;

list <int> addpoint;

for (iter = point.begin(); iter != point.end(); iter++)

{

if (\*iter % 2 != 0)

{

addpoint.push\_back(\*iter);

}

}

point.clear();

point.insert(iter, addpoint.begin(), addpoint.end());

copy(point.begin(), point.end(), ostream\_iterator<int>(cout, " "));

return 0;

}

**Скриншоты**

