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

Вариант 5

Выполнил работу

студент группы РИС-20-1б

Зверев А.Д.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

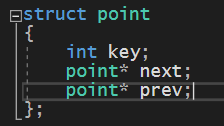
Пермь, 2021

**Постановка задачи**

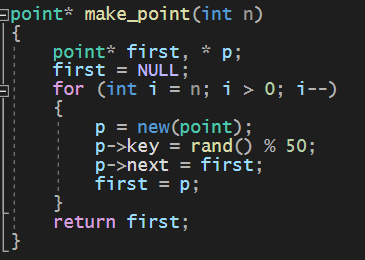
1. Сформировать двунаправленный список типа int.
2. Распечатать полученную структуру.
3. Удалить из списка все четные элементы.
4. Распечатать полученный результат.

**Анализ задачи**

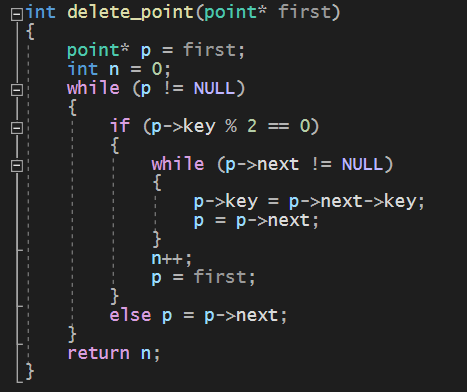
1. Для решения задачи необходимо…
   1. Создать структуру point, которая содержит в себе данные типа int и указатель на следующий элемент одностороннего списка;



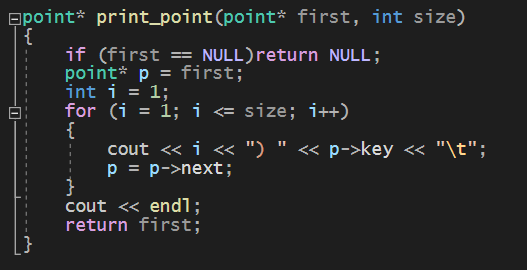
* 1. Написать функцию make\_point, которая будет давать значение элементам списка;



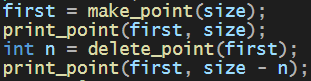
* 1. Написать функцию delete\_point, которая будет удалять все четные элементы;



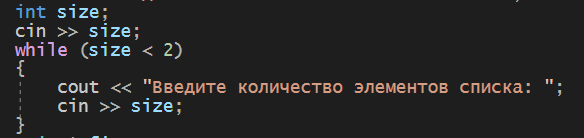
* 1. Написать функцию print\_point, которая выводит элементы списка в консоль;



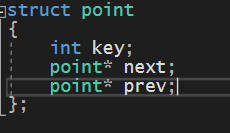
* 1. Вызвать все вышеперечисленные функции в основном теле программы;



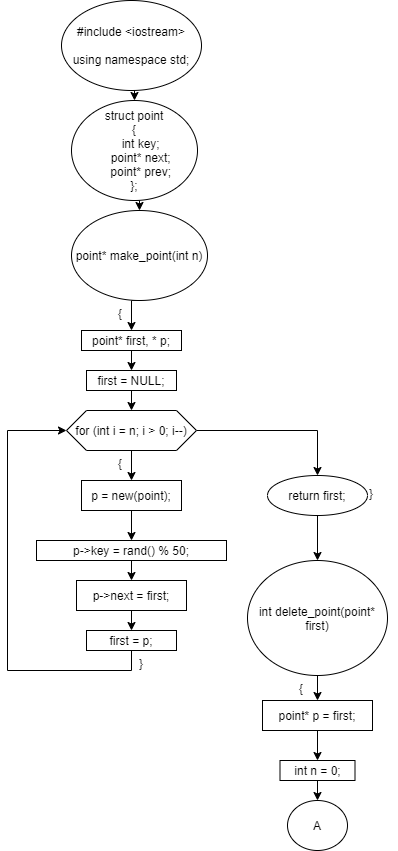
1. Для решения задачи были использованы данные следующих типов:
   1. Int для хранения количества элементов списка;

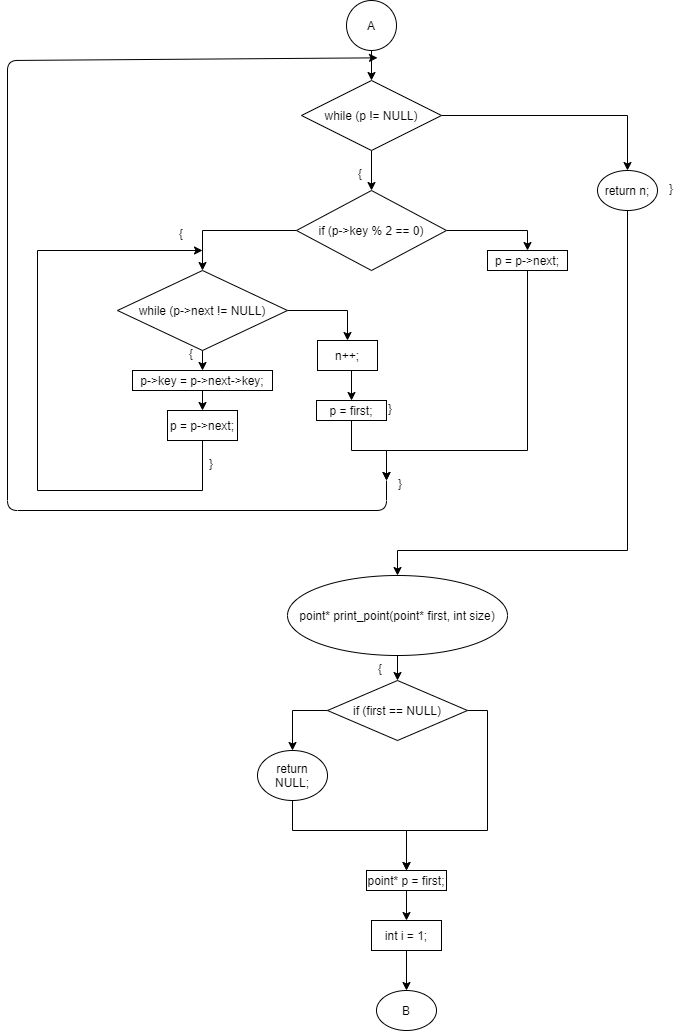
****

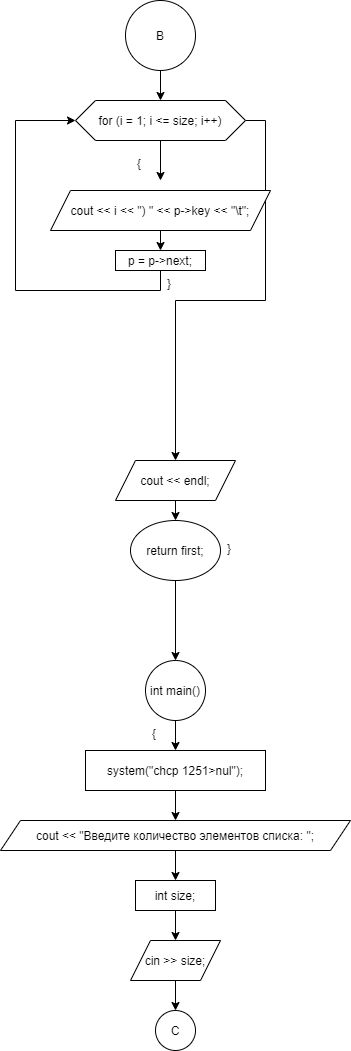
* 1. Структура для хранения двунаправленного списка;

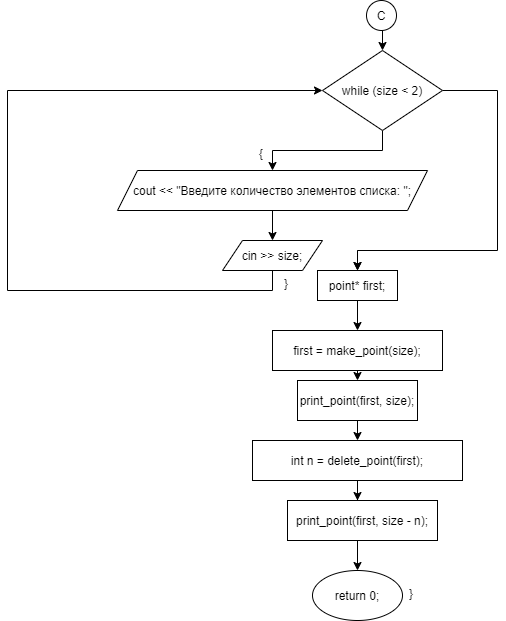


**Блок—схема программы**









**Решение**

#include <iostream>

using namespace std;

struct point

{

int key;

point\* next;

point\* prev;

};

point\* make\_point(int n)

{

point\* first, \* p;

first = NULL;

for (int i = n; i > 0; i--)

{

p = new(point);

p->key = rand() % 50;

p->next = first;

first = p;

}

return first;

}

int delete\_point(point\* first)

{

point\* p = first;

int n = 0;

while (p != NULL)

{

if (p->key % 2 == 0)

{

while (p->next != NULL)

{

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

p = p->next;

}

n++;

p = first;

}

else p = p->next;

}

return n;

}

point\* print\_point(point\* first, int size)

{

if (first == NULL)return NULL;

point\* p = first;

int i = 1;

for (i = 1; i <= size; i++)

{

cout << i << ") " << p->key << "\t";

p = p->next;

}

cout << endl;

return first;

}

int main()

{

system("chcp 1251>nul");

cout << "Введите количество элементов списка: ";

int size;

cin >> size;

while (size < 2)

{

cout << "Введите количество элементов списка: ";

cin >> size;

}

point\* first;

first = make\_point(size);

print\_point(first, size);

int n = delete\_point(first);

print\_point(first, size - n);

return 0;

}

**Скриншоты**

