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 №11.2**

Дисциплина: основы алгоритмизации и программирования

Тема: “Динамические структуры данных” в Си

Выполнил работу

студент группы ИВТ-20-2б

Солдатов А.П.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

****Цель работы****

Цель – разработать программу для выполнения операций с динамическими структурами данных.

Для достижения цели нужно решить следующие задачи:

* Найти как создаются и объявляются динамические структуры данных.
* Найти как работают динамические структуры данных.
* Найти как выполняется операция добавление элемента.
* Разработать программу.

Постановка задачи

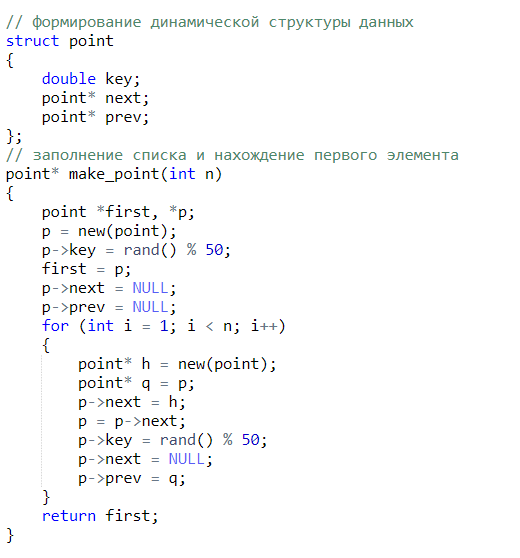
Сформировать двунаправленный массив с типом информационного поля int. Вывести структуру в консоль. Удалить из списка первый элемент с четным информационным полем. Вывести структуру в консоль.

Для выполнения данной работы использован язык программирования C++ в программной среде Visual Studio 2019.

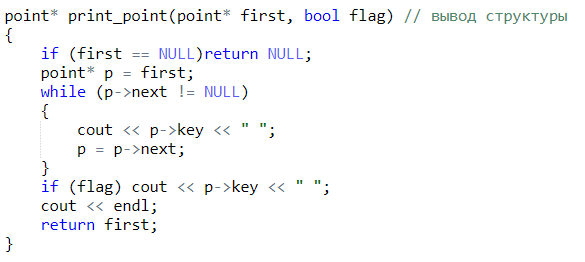
Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

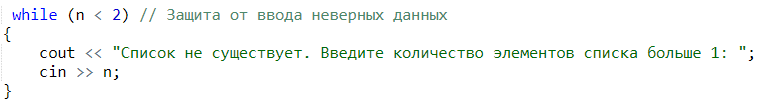
* Формирование динамической структуры данных



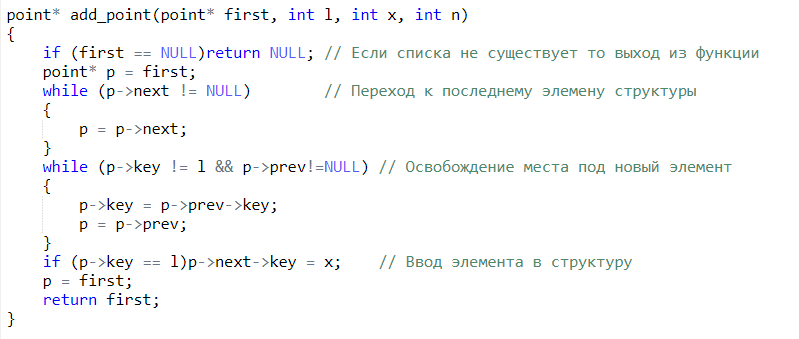
* Вывод данных на экран.



* Защита от ввода неверных данных:

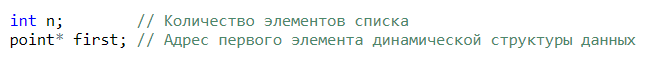


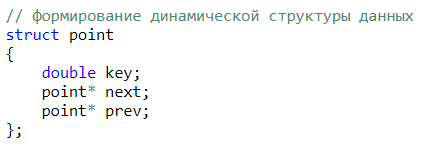
* Добавление элемента после элемента с заданным информационным полем:



Flag является глобальной переменной, из-за чего срабатывает в функции вывода.

1. Какой тип понадобиться для работы:









А также i для счетчика.

1. В каком виде эти данные будут представлены для решения задачи:

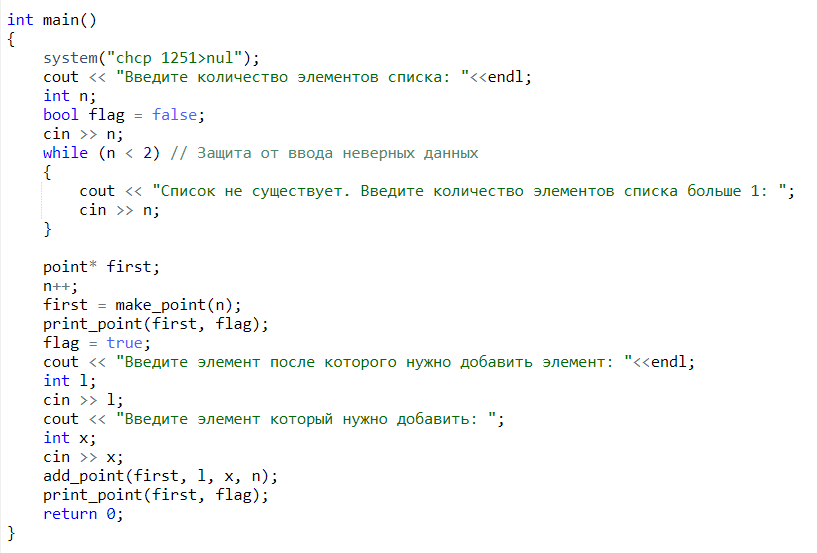
Ввод в консоль количество элементов в списке.

Заполняется динамическая структура данных с помощью функции rand().

1. Какими операторами будет реализован ввод и вывод данных:

Ввод-вывод на экран осуществляется при помощи операторов cin и cout соответственно.

Выполнение действий (вызов соответствующих подпрограмм) осуществляется в главной программе.



Код

#include <iostream>

using namespace std;

struct point

{

double key;

point\* next;

point\* prev;

};

point\* make\_point(int n)

{

point \*first, \*p;

p = new(point);

p->key = rand() % 50;

first = p;

p->next = NULL;

p->prev = NULL;

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

{

point\* h = new(point);

point\* q = p;

p->next = h;

p = p->next;

p->key = rand() % 50;

p->next = NULL;

p->prev = q;

}

return first;

}

point\* add\_point(point\* first, int l, int x, int n)

{

if (first == NULL)return NULL;

point\* p = first;

while (p->next != NULL)

{

p = p->next;

}

while (p->key != l && p->prev!=NULL)

{

p->key = p->prev->key;

p = p->prev;

}

if (p->key == l)p->next->key = x;

p = first;

return first;

}

point\* print\_point(point\* first, bool flag) // вывод структуры

{

if (first == NULL)return NULL;

point\* p = first;

while (p->next != NULL)

{

cout << p->key << " ";

p = p->next;

}

if (flag) cout << p->key << " ";

cout << endl;

return first;

}

int main()

{

system("chcp 1251>nul");

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

int n;

bool flag = false;

cin >> n;

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

{

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

cin >> n;

}

point\* first;

n++;

first = make\_point(n);

print\_point(first, flag);

flag = true;

cout << "Введите элемент после которого нужно добавить элемент: "<<endl;

int l;

cin >> l;

cout << "Введите элемент который нужно добавить: ";

int x;

cin >> x;

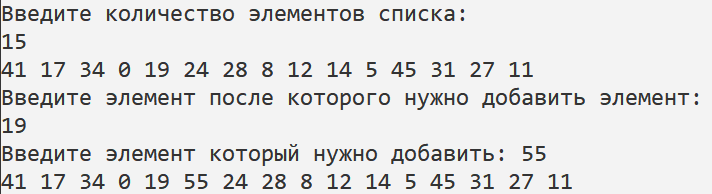
add\_point(first, l, x, n);

print\_point(first, flag);

return 0;

}

Скриншот выполнения программы



Блок-схема

