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 №11.3**

Дисциплина: «основы алгоритмизации и программирования»

Тема: Информационные и динамические структуры. Стеки

Вариант 5

Выполнил работу

студент группы РИС-20-1б

Зверев А.Д.

Проверила

Доцент кафедры ИТАС

Полякова О.А.

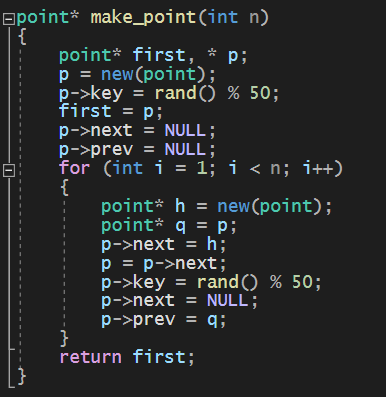
Пермь, 2021

**Постановка задачи**

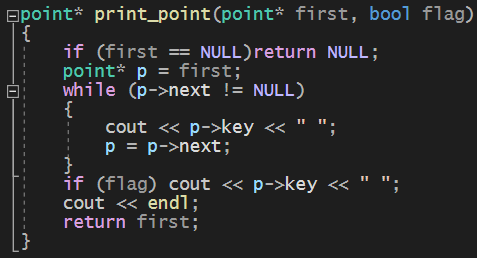
1. Сформировать информационное поле типа char\*.
2. Распечатать полученную структуру.
3. Добавить в стек элемент после элемента с заданным информационным полем.
4. Распечатать полученный результат.

**Анализ задачи**

1. Для решения задачи необходимо…
   1. Создать функцию make\_point, которая получает на ввод число n, по которому и формирует стек с заданным n количеством чисел.



* 1. Реализовать функцию print\_point, которая выводит элементы стека в консоль;

****

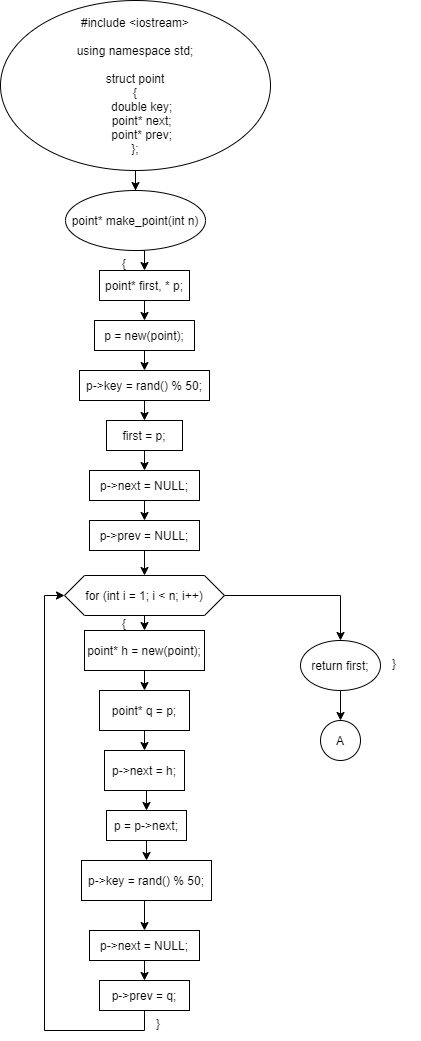
1. В программе были использованы следующие типы данных:
   1. Тип данных int, для хранения количества элементов;

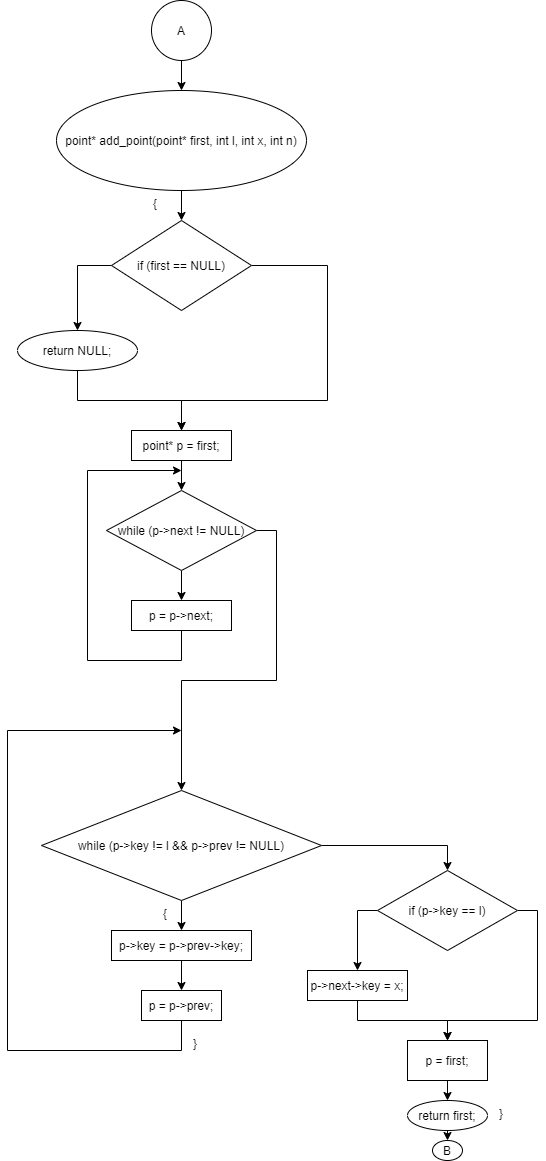


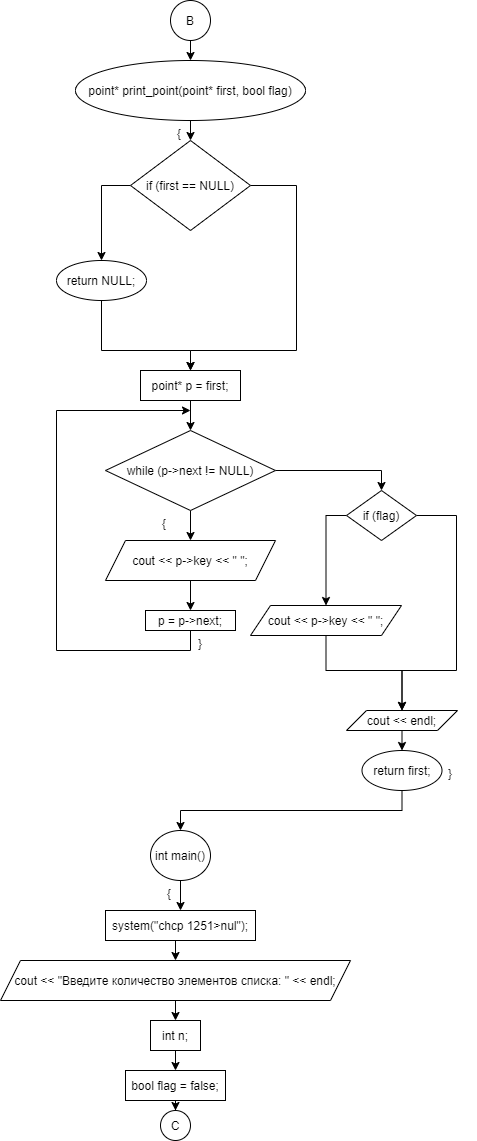
* 1. Тип данных bool для логического флажка, проверяющего правильность ввода данных и вывода стека в консоль;

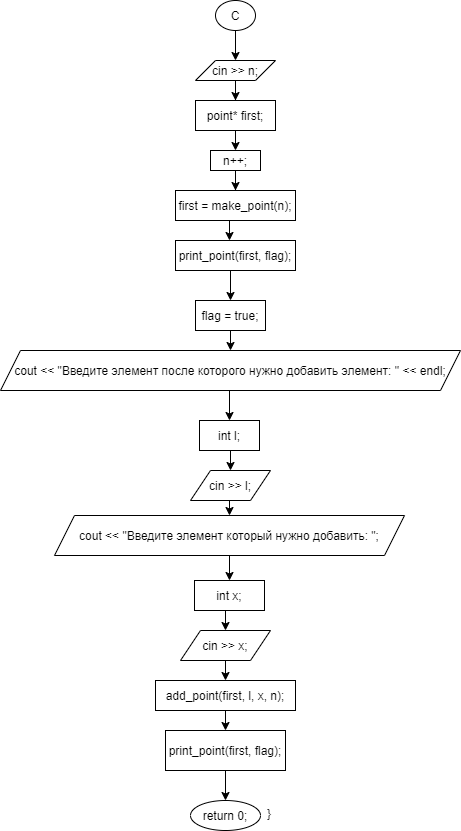


**Блок—схема программы**









**Решение**

#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;

}

**Скриншоты**

