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 №14**

Дисциплина: «Основы теории алгоритмов и структуры данных»

Тема: Алгоритмы поиска. Работа со списками.

Метод интерполяционного поиска.

Вариант 17

Выполнил:

Студент группы РИС-20-1б

Томилов Владислав Алексеевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

1)  Научиться выполнять поиск элемента в массиве по ключу.

**Постановка задачи**

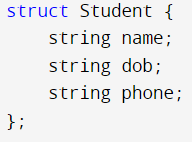
(Вариант 23)

1. Создать динамический массив из записей (в соответствии с вариантом), содержащий не менее 100 элементов. Для заполнения элементов массива использовать ДСЧ.
2. Предусмотреть сохранение массива в файл и загрузку массива из файла.
3. Предусмотреть возможность добавления и удаления элементов из массива (файла).
4. Выполнить поиск элемента в массиве по ключу в соответствии с вариантом. Для поиска использовать метод линейного поиска, метод Прямого поиска подстроки в строке и интерполяционный метод.

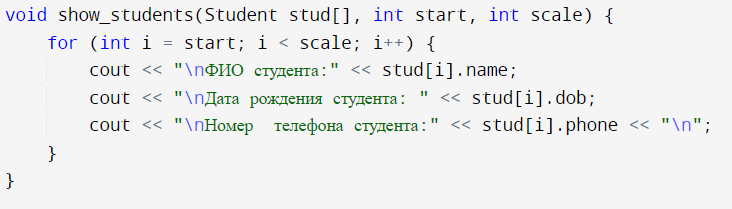
**Анализ задачи**

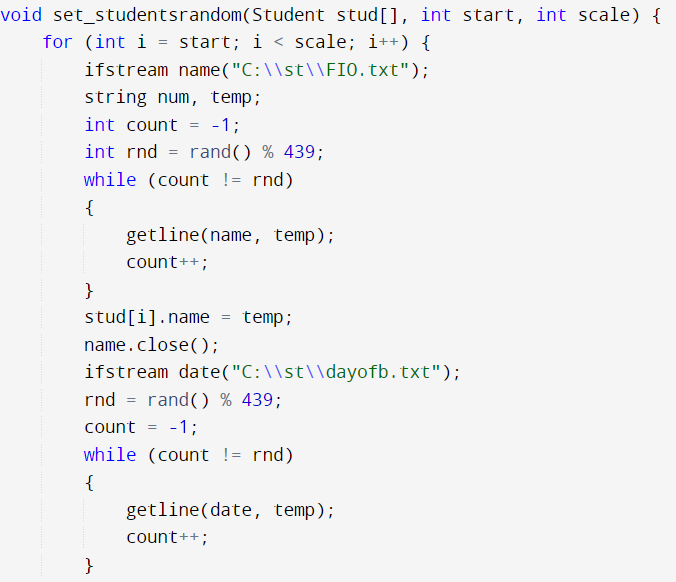
**1.** Для решения задачи необходимо:

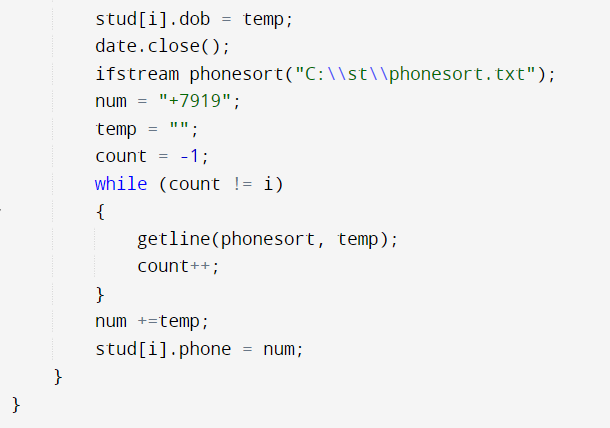
**1.1.** Создать структуру для реализации поиска.



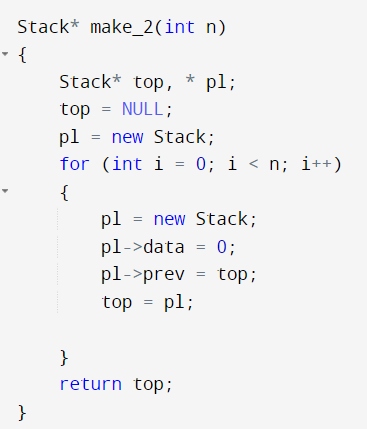
**1.2.** Разработать функцию “show\_studetns” для вывода созданного массива структур.

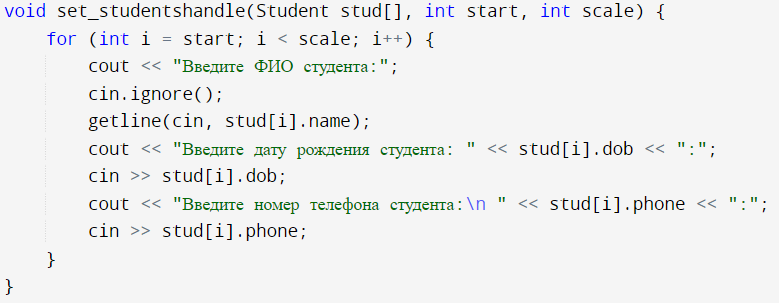


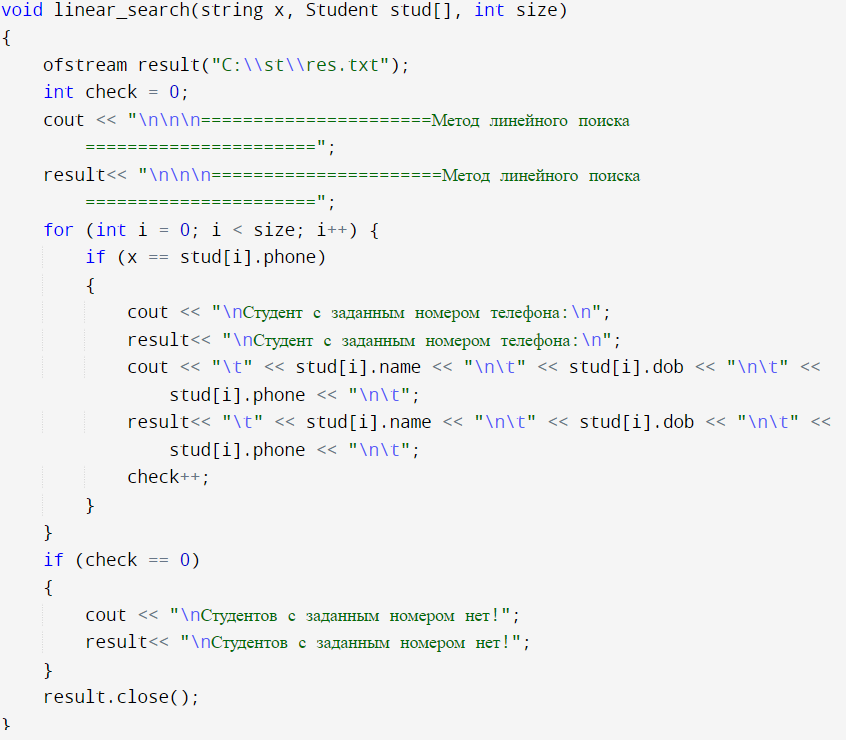
**1.3.** Разработать функцию set\_studentsrandom, которая будет заполнять структуру данными из какого-то заранее заготовленного файла, например FIO.txt



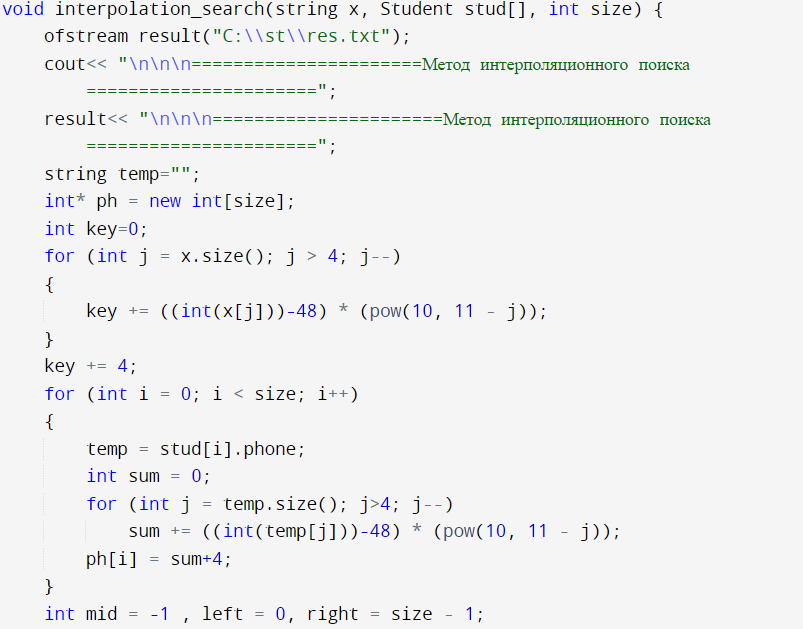
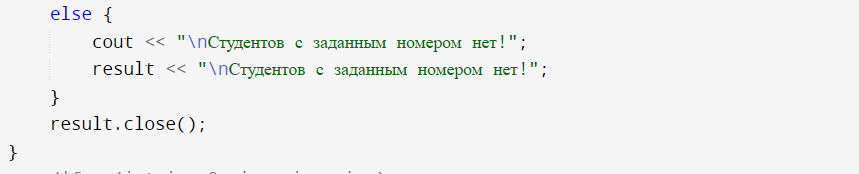
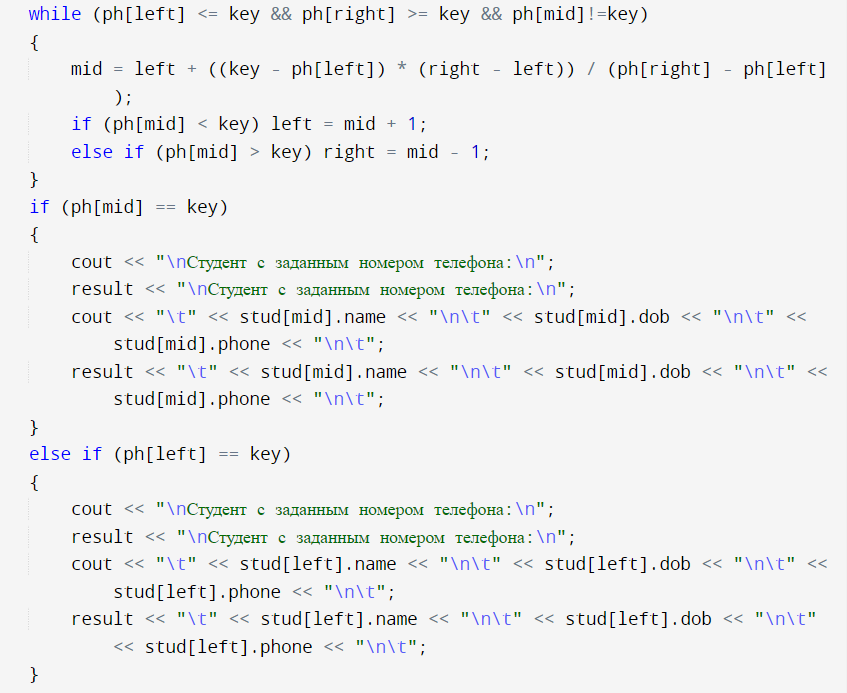
**1.4.** Разработать функцию set\_studentshandle для ручного ввода данных о студентах.



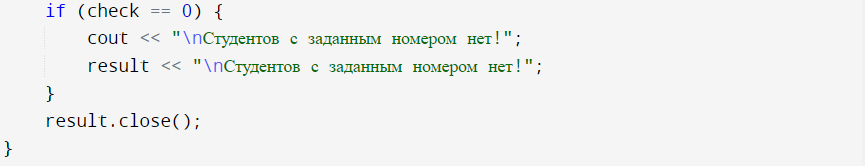
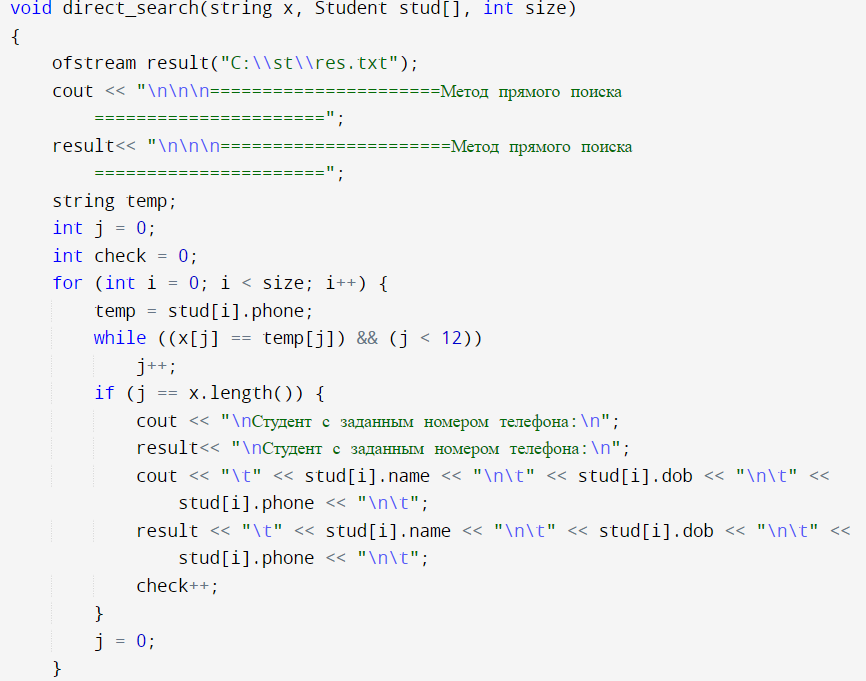


**1.5.** Разработать функцию linear\_search, которая производит линейный поиск в массиве структур. 

**1.6.** Разработать функцию interpolation\_search, которая производит линейный поиск в массиве структур.

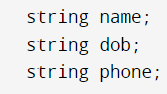


**1.7.** Разработать функцию direct\_search, которая производит прямой поиск в массиве структур.



**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Структура имеет информационные поля типа string.

****

**2.2.** Для функции set\_studentsrandom, используются следующие аргументы:

****

**2.4.** Для функции set\_studentshandle, используются следующие аргументы:



**2.5.** Для функции show\_students используются следующие аргументы:



**2.6.** Для функции direct\_search используются следующие аргументы:

****

**2.7.** Для функции linear\_search используются следующие аргументы:



**2.8.** Для функции interpolation\_search используются следующие аргументы:

****

**3.** Для решения задачи данные были представлены в следующем виде:

**3.1.** Для заполнения массива структур пользователю было предложено два варианта ввода: ввод с клавиатуры или заполнение значениями из файла.

**4.** Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.** Ввод данных реализован с помощью оператора cin.



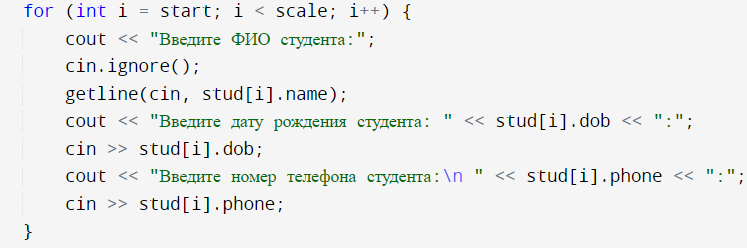
**4.2.** Вывод данных на консоль реализован с помощью оператора cout.



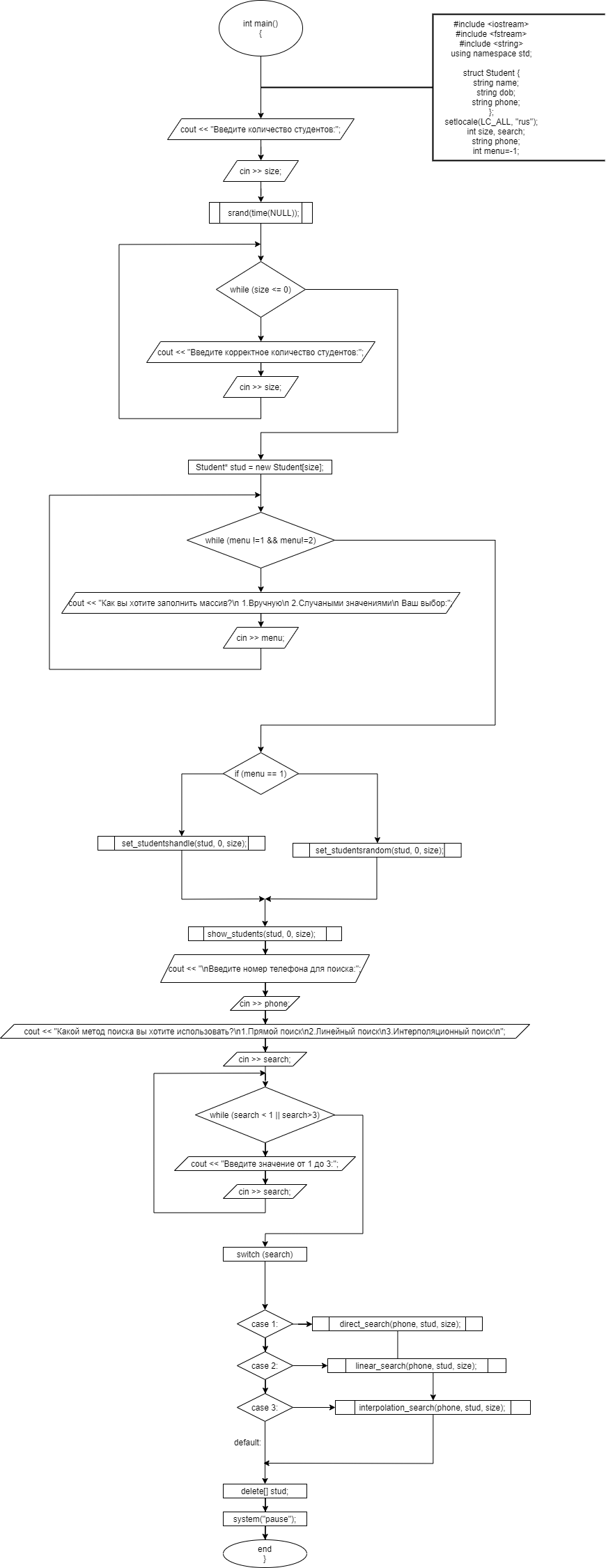
**5.** Поставленные задачи будут решены следующими действиями:

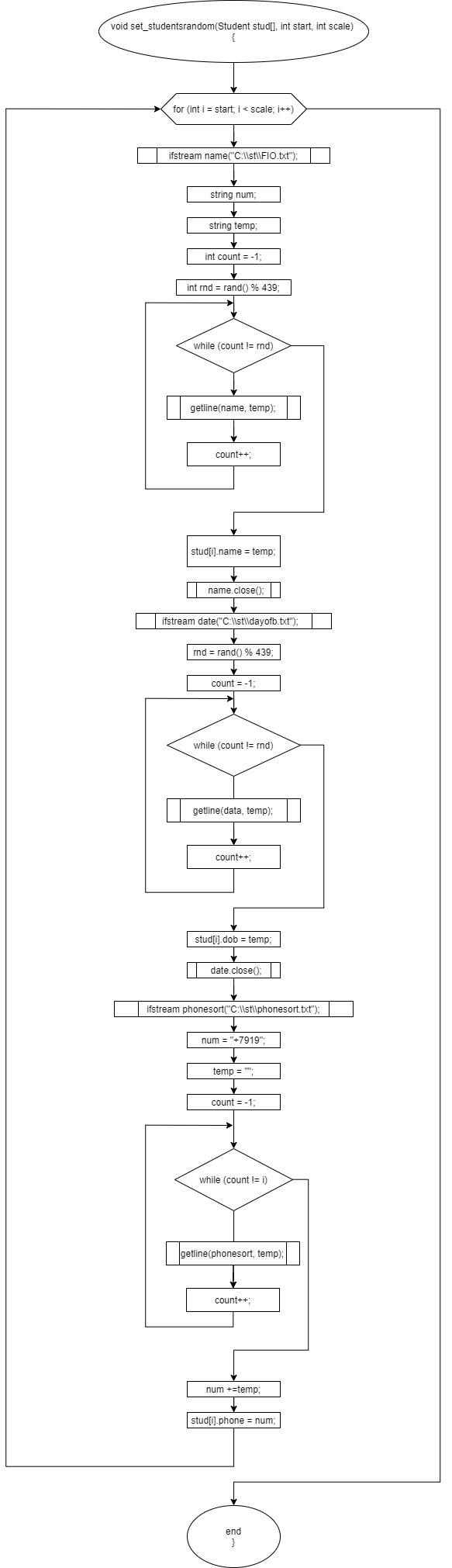
**5.1.** Для заполнения массива используется цикл for:

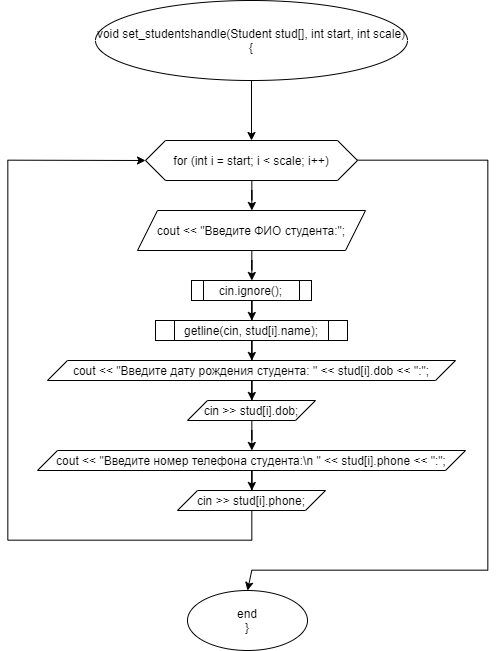


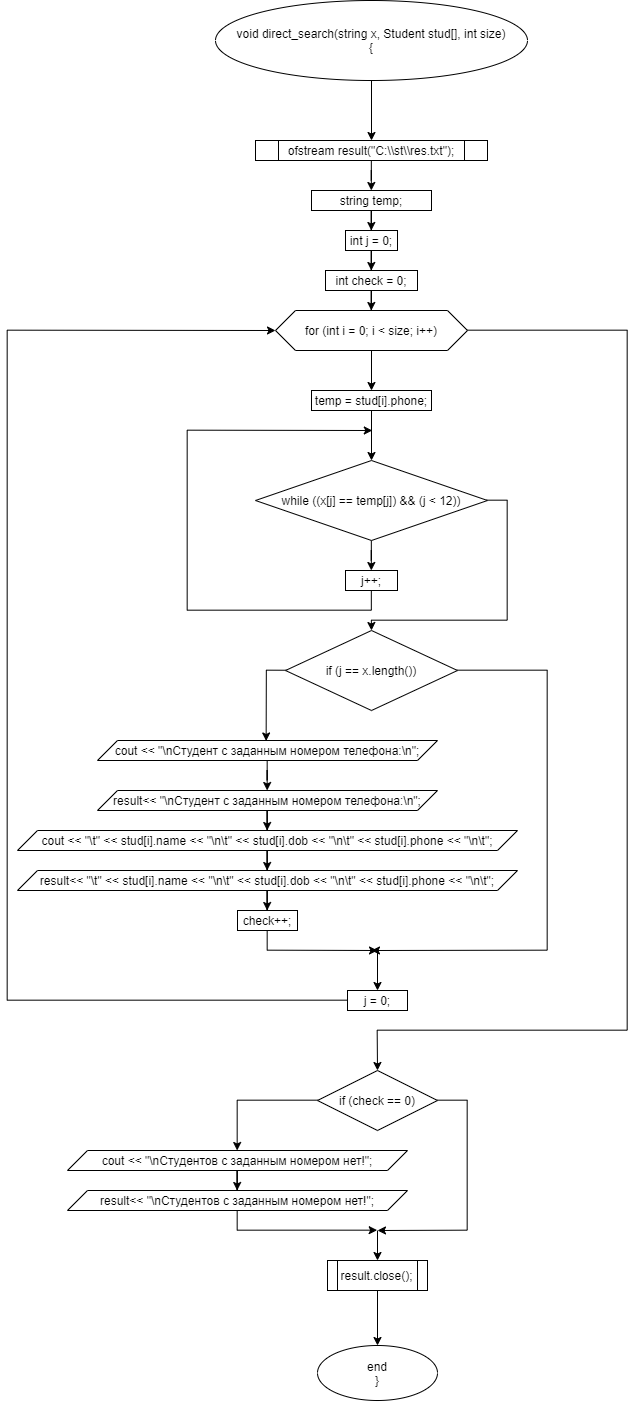
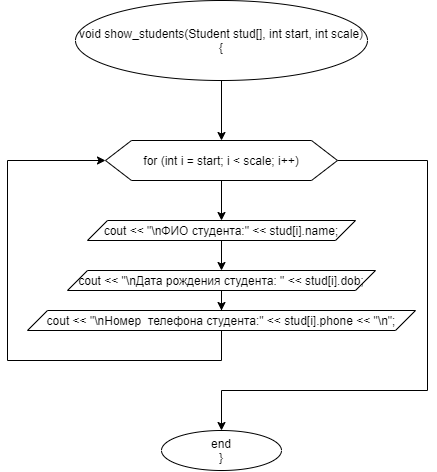


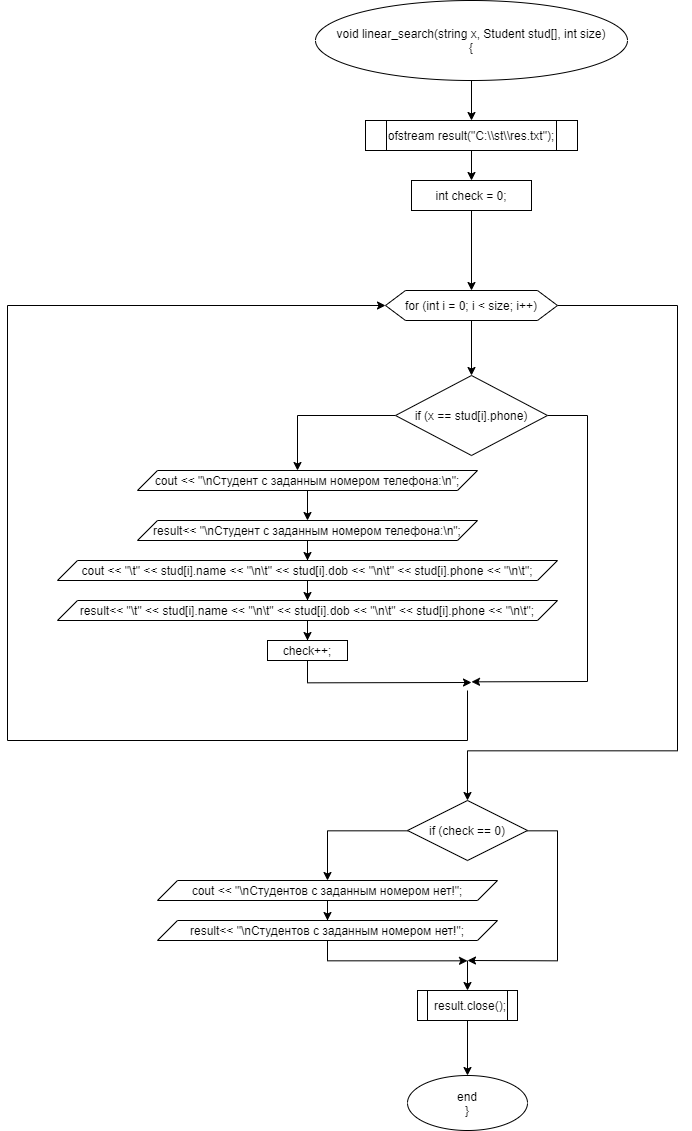
**Блок-схема**

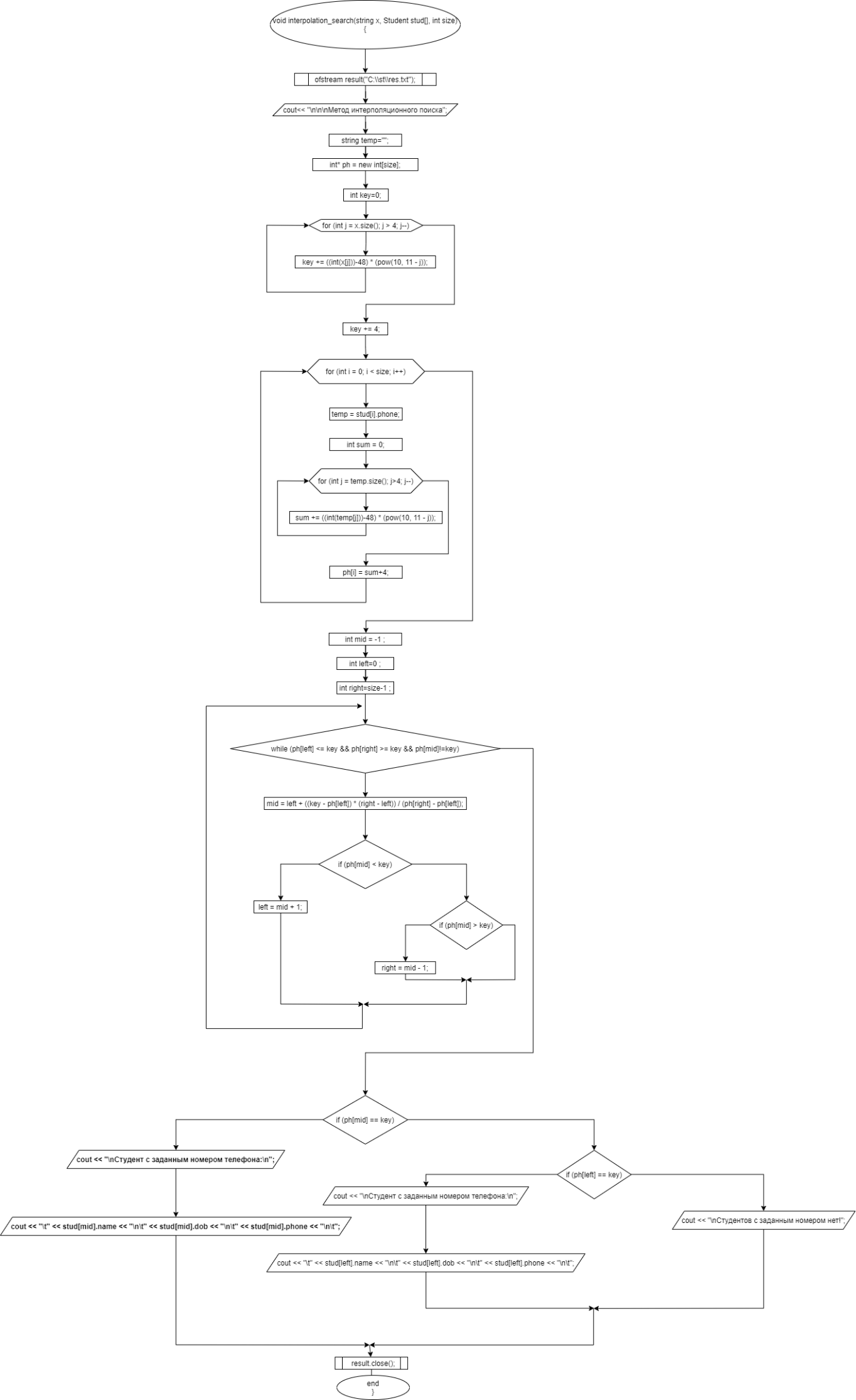


****









**Код программы**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct Student {

string name;

string dob;

string phone;

};

void set\_studentsrandom(Student stud[], int start, int scale) {

int tempphone = 1000000;

for (int i = start; i < scale; i++) {

ifstream name("C:\\Users\\Влад\\Desktop\\c++(2)\\lab12\\FIO.txt");

string num, temp;

int count = -1;

int rnd = rand() % 439;

while (count != rnd)

{

getline(name, temp);

count++;

}

stud[i].name = temp;

name.close();

ifstream date("C:\\Users\\Влад\\Desktop\\c++(2)\\lab12\\DOB.txt");

rnd = rand() % 439;

count = -1;

while (count != rnd)

{

getline(date, temp);

count++;

}

stud[i].dob = temp;

date.close();

num = "+7919";

num += to\_string(tempphone);

tempphone++;

stud[i].phone = num;

}

}

void set\_studentshandle(Student stud[], int start, int scale) {

for (int i = start; i < scale; i++) {

cout << "Введите ФИО студента:";

cin.ignore();

getline(cin, stud[i].name);

cout << "Введите дату рождения студента: " << stud[i].dob << ":";

cin >> stud[i].dob;

cout << "Введите номер телефона студента:\n " << stud[i].phone << ":";

cin >> stud[i].phone;

}

}

void show\_students(Student stud[], int start, int scale) {

for (int i = start; i < scale; i++) {

cout << "\nФИО студента:" << stud[i].name;

cout << "\nДата рождения студента: " << stud[i].dob;

cout << "\nНомер телефона студента:" << stud[i].phone << "\n";

}

}

void direct\_search(string x, Student stud[], int size)

{

cout << "\n\n\n======================Метод прямого поиска======================";

string temp;

int j = 0;

int check = 0;

for (int i = 0; i < size; i++) {

temp = stud[i].phone;

while ((x[j] == temp[j]) && (j < 12))

j++;

if (j == x.length()) {

cout << "\nСтудент с заданным номером телефона:\n";

cout << "\t" << stud[i].name << "\n\t" << stud[i].dob << "\n\t" << stud[i].phone << "\n\t";

check++;

}

j = 0;

}

if (check == 0)

cout << "\nСтудентов с заданным номером нет!";

}

void linear\_search(string x, Student stud[], int size)

{

int check = 0;

cout << "\n\n\n======================Метод линейного поиска======================";

for (int i = 0; i < size; i++) {

if (x == stud[i].phone)

{

cout << "\nСтудент с заданным номером телефона:\n";

cout << "\t" << stud[i].name << "\n\t" << stud[i].dob << "\n\t" << stud[i].phone << "\n\t";

check++;

}

}

if (check == 0)

cout << "\nСтудентов с заданным номером нет!";

}

void interpolation\_search(string x, Student stud[], int size) {

cout << "\n\n\n======================Метод интерполяционного поиска======================"<<endl;

string temp = "";

int\* ph = new int[size];

int key = 0;

for (int j = x.size(); j > 4; j--)

{

key += ((int(x[j])) - 48) \* (pow(10, 11 - j));

}

key += 4;

//cout << "\nКлюч= " << key << '\n';

for (int i = 0; i < size; i++)

{

temp = stud[i].phone;

int sum = 0;

for (int j = temp.size(); j > 4; j--)

sum += ((int(temp[j])) - 48) \* (pow(10, 11 - j));

ph[i] = sum + 4;

}

int mid = -1, left = 0, right = size - 1;

while (ph[left] <= key && ph[right] >= key && ph[mid] != key)

{

mid = left + ((key - ph[left]) \* (right - left)) / (ph[right] - ph[left]);

if (ph[mid] < key) left = mid + 1;

else if (ph[mid] > key) right = mid - 1;

}

if (ph[mid] == key)

{

cout << "\nСтудент с заданным номером телефона:\n";

cout << "\t" << stud[mid].name << "\n\t" << stud[mid].dob << "\n\t" << stud[mid].phone << "\n\t";

}

else if (ph[left] == key)

{

cout << "\nСтудент с заданным номером телефона:\n";

cout << "\t" << stud[left].name << "\n\t" << stud[left].dob << "\n\t" << stud[left].phone << "\n\t";

}

else cout << "Студентов с заданным номером нет!"<<endl;

}

/\*for (int i = 0; i < size; i++)

{

cout << ph[i] << '\n';

}\*/

int main()

{

system("chcp 1251>nul");

setlocale(LC\_ALL, "rus");

int size;

string phone;

int menu;

cout << "Введите количество студентов:";

while (!(cin >> size) || (cin.peek() != '\n') || size <= 0)

{

cin.clear();

while (cin.get() != '\n');

{

cout << "Введите количество студентов:";

}

}

cout << endl;

Student\* stud = new Student[size];

cout << "Как вы хотите заполнить массив?\n 1.Вручную\n 2.Случаными значениями\n Ваш выбор:";

cin >> menu;

if (menu == 1)

set\_studentshandle(stud, 0, size);

else set\_studentsrandom(stud, 0, size);

show\_students(stud, 0, size);

cout << "\nВведите номер телефона для поиска:";

cin >> phone;

direct\_search(phone, stud, size);

linear\_search(phone, stud, size);

interpolation\_search(phone, stud, size);

delete[] stud;

system("pause");

**Результат работы программы**

