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 №12**

Дисциплина: «Основы теории алгоритмов и структуры данных»

Тема: Алгоритмы поиска. Работа со списками. Метод интерполяционного поиска

Вариант 1

Выполнила:

Студент группы РИС-20-1б

Азмагулов Артём Вадимович

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2021**

**Цель работы**

Работа с динамическими структурами данных.

**Постановка задачи**

Задание:

1. Создать динамический массив из записей (в соответствии с вариантом), содержащий не менее 30 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).

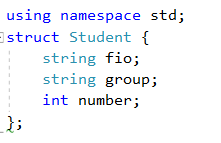
4. Реализовать поиски данных линейным, интерполяционным и прямым поисками.

|  |  |  |
| --- | --- | --- |
| **Вариант** | **Данные** | **Ключ (string)** |
| 1 | ФИО, группа, рейтинг | ФИО |

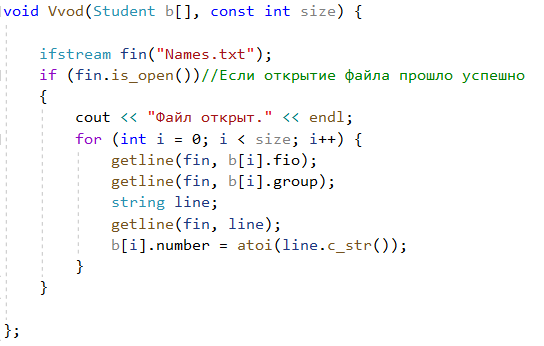
**Анализ задачи**

**1.** Для решения задачи необходимо:

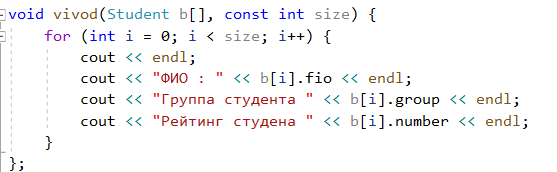
**1.1. Создать структуру данных с типами данных string, string, int:**



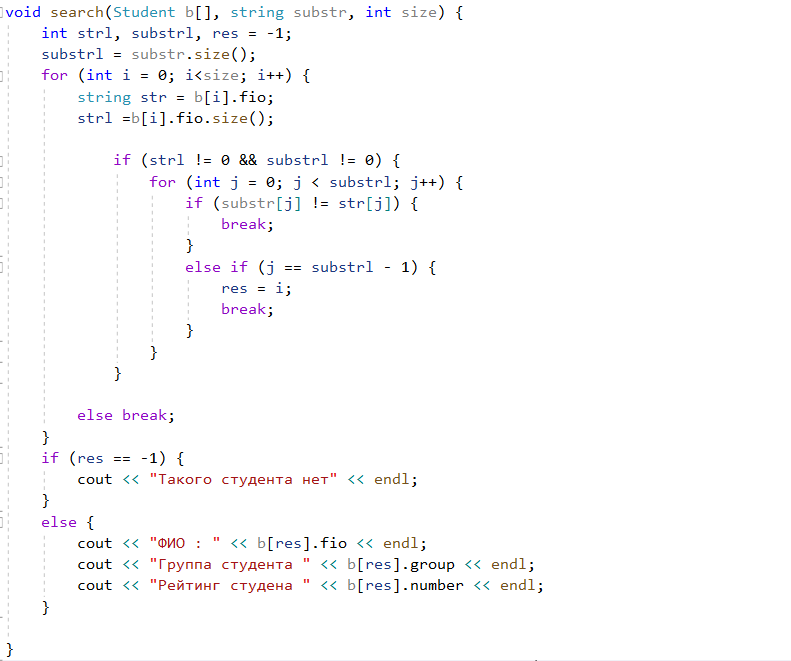
**1.2.** Разработать функцию для заполнения структуры через цикл for, из файла:



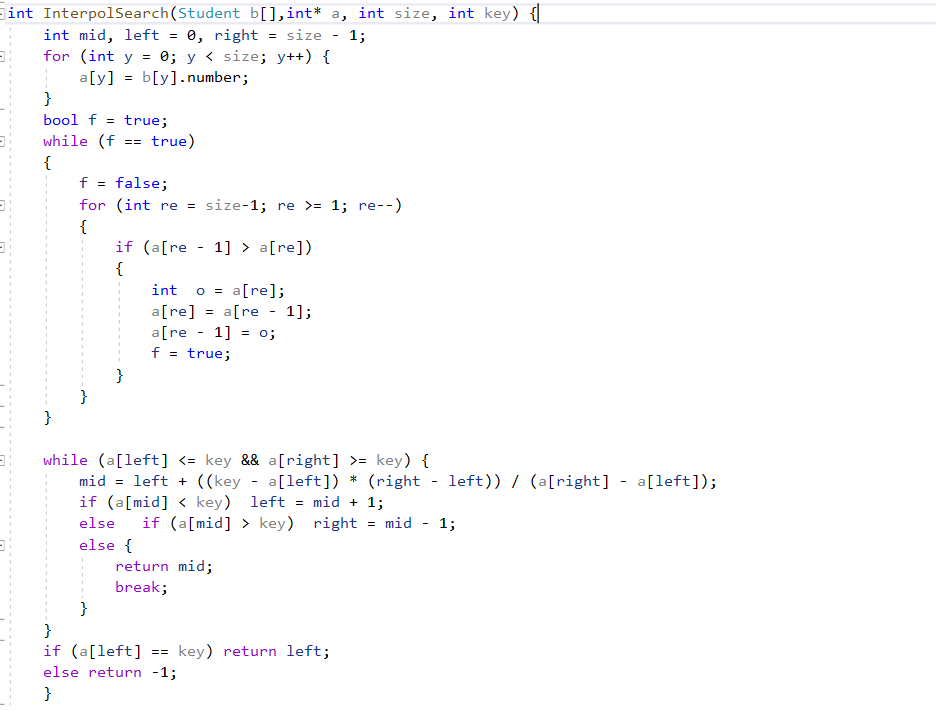
**1.3.Разработать функцию вывода всех полученных программой значений структур:**



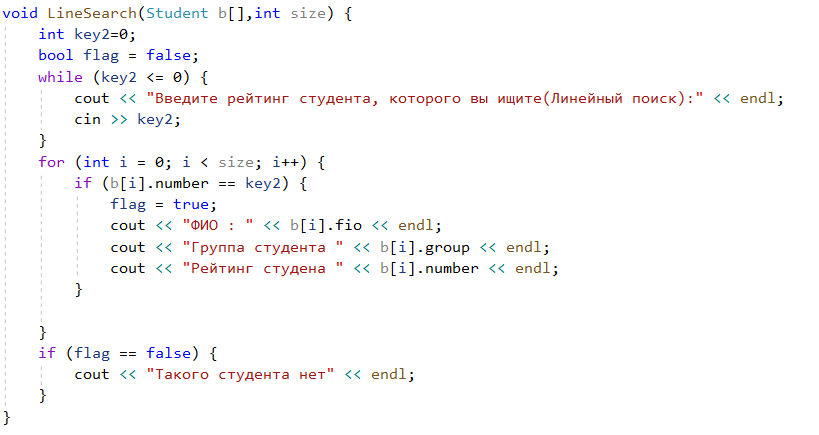
**1.4.** Реализовать прямой поиск подстроки в строке, опираясь на функцию for с дальнейшим перебором всех элементов структуры:



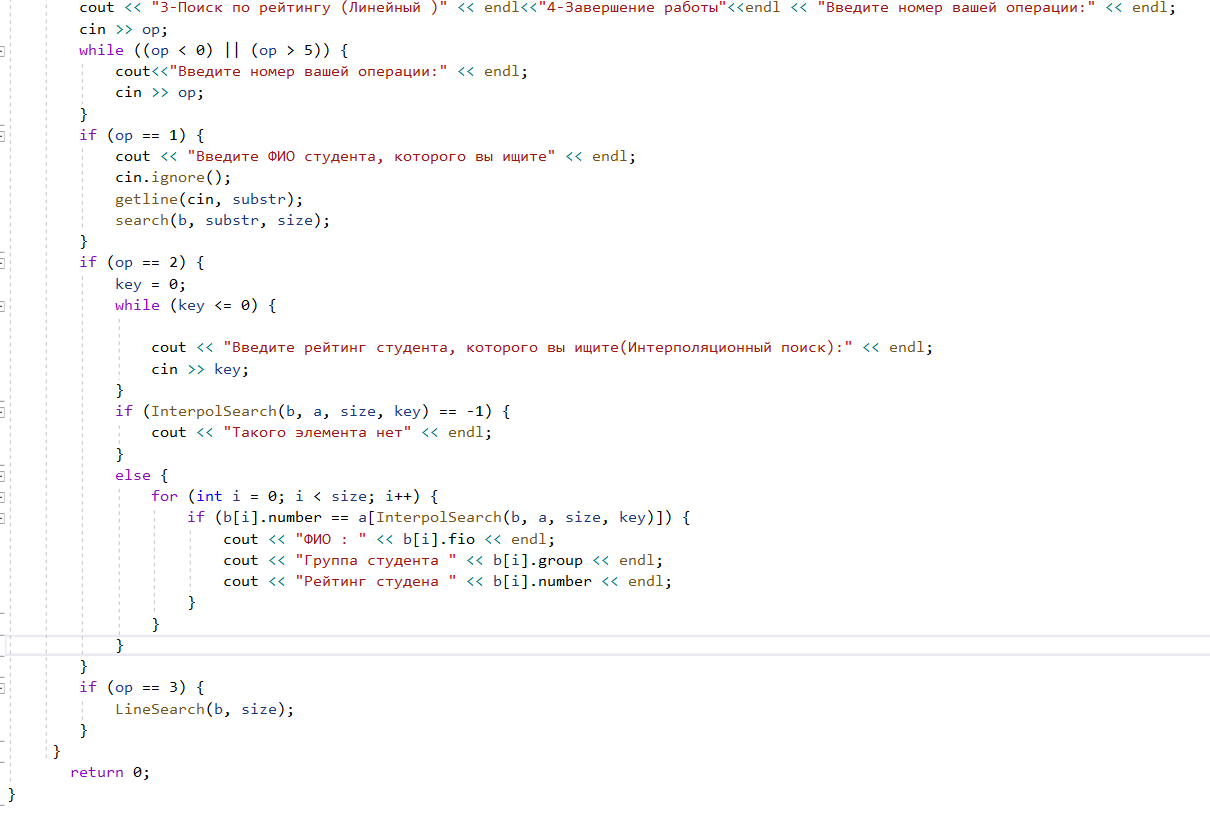
**1.5.** Реализовать интерполяционный поиск по рейтингу студентов, опираясь на цикл for, преждевременно отсортировав элементы методом обмена (Пузырька):



**1.6.** Реализовать линейный поиск по рейтингу студентов, постепенно переходя к сравнению элементов с заданным ключом используя цикл for:



**1.6.** Реализовать меню, с доступом ко всем выше перечисленным функциям, в том числе к функции с завершением работы:



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

**2.1.**Для работы функции Vvod принимается обращение к элементам структуры и константному значению размера структуры:



**2.2.** Для работы функции vivod принимается обращение к элементам структуры и константному значению размера структуры(const int):



**2.3.** Для работы функции search принимается обращение к элементам структуры, константному значению размера структуры(const int) и размер символьного ключа искомого элемента(int):



**2.4.** Для работы функции InterpolSearch принимается обращение к элементам структуры, константному значению размера структуры(const int), указателю на массив элементов , полученных из ранее объявленной структуры, и значение искомого ключа(int):



**2.5.** Для работы функции LineSearch принимается обращение к элементам структуры, константному значению размера структуры(const int):



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

**3.1.** Для обработки строк из файла данные были записаны в виде структуры Student, содержащий в себе строковые значения ФИО, группы и целочисленного рейтинга студента.

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

**4.1.**Ввод данных из файла "Names.txt" реализован с помощью функции getline.

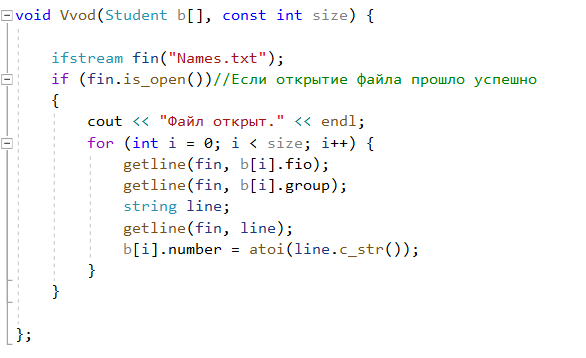


**4.2.** Вывод данных на консоль реализован с помощью оператора cout.

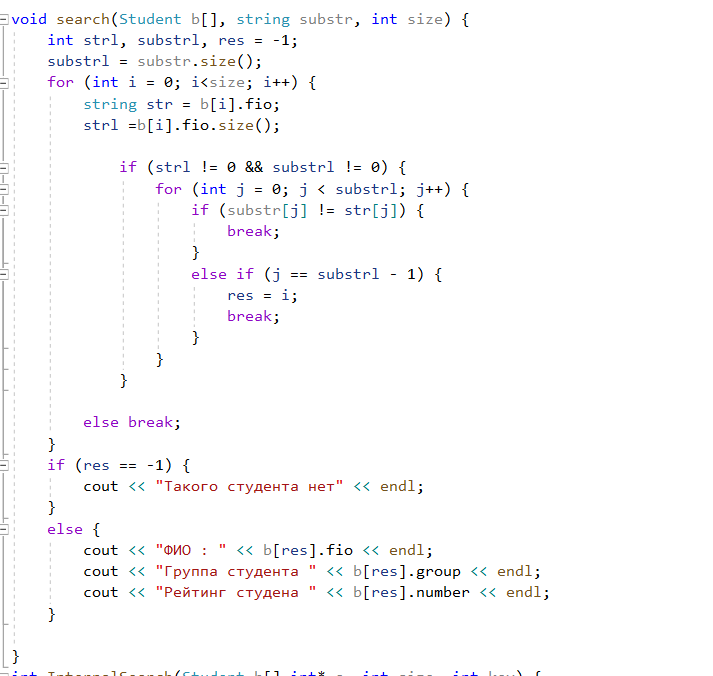


**5.**Поставленные задачи будут решены следующими действиями:

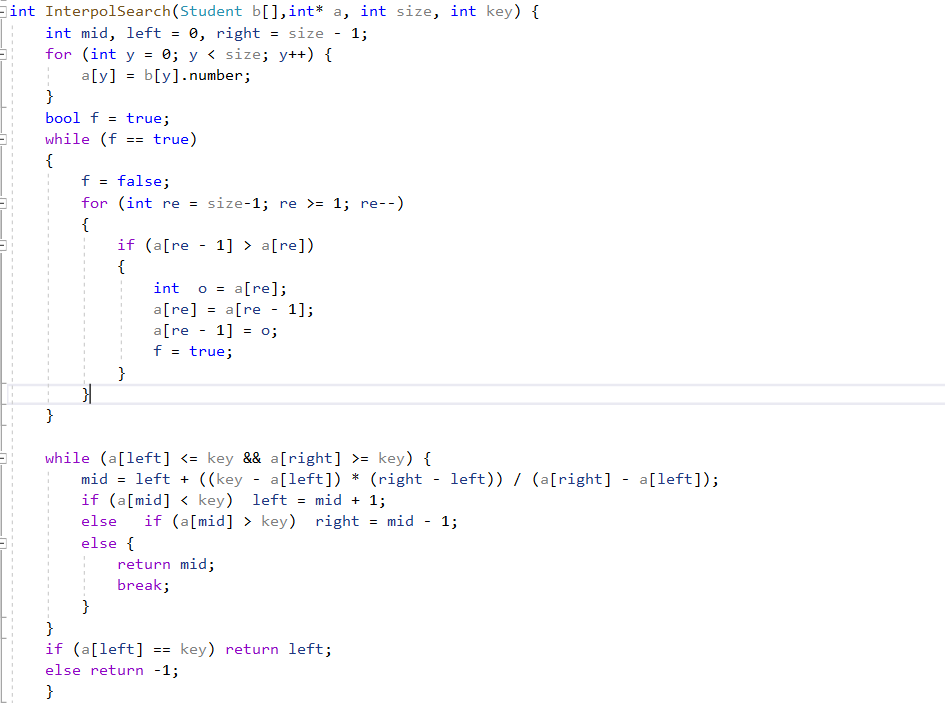
**5.1.** Для работы с информацией из файла построчно, все операции со строкой проводятся внутри цикла for, после каждого полного прохождения тела цикла производится считывание следующей строки. Повторение цикла продолжается, пока не достигнет конца файла.



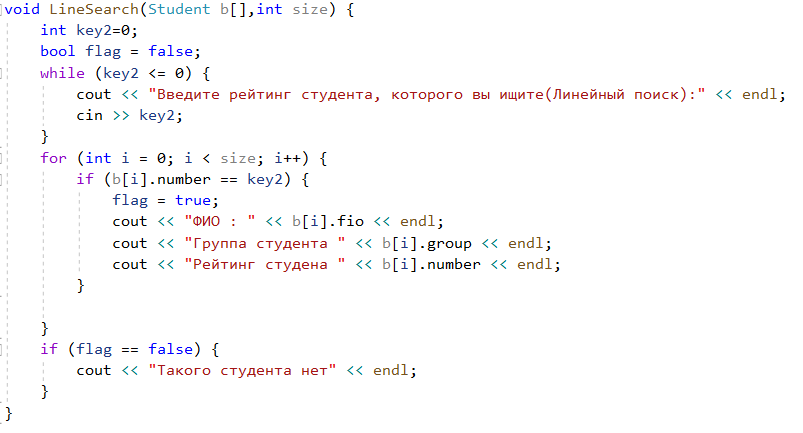
**5.2.**Поиск подстроки в строке происходит после приема ключа функцией, в дальнейшем которая проходит пошаговый перебор всех элементов структуры с последующим сравнением символом, с условием вывода местоположения элемента в случае лишь достижения конца подстроки:



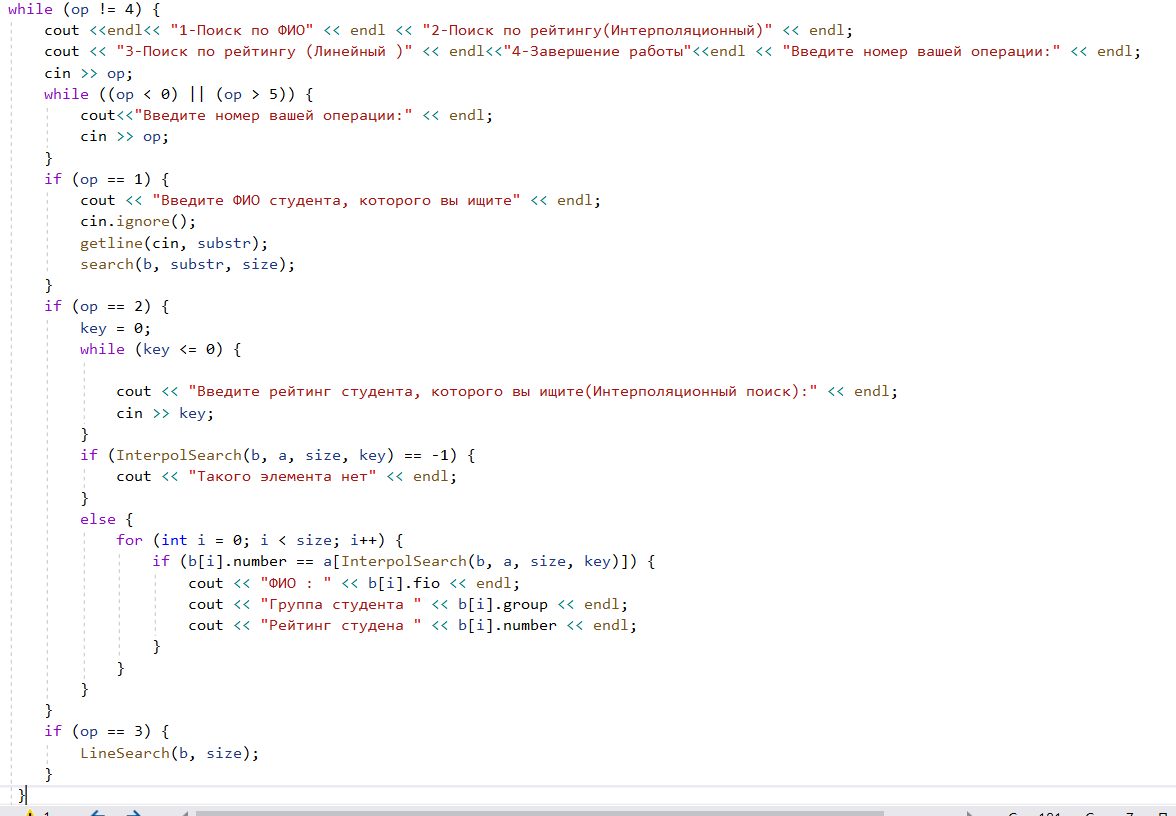
**5.3. .**Интерполяционный поиск элемента происходит после приема ключа функцией, в дальнейшем которая создает массив из элементов структуры, расставляя их по ходу приема в порядке возрастания. Затем следует приближенный поиск нужного элемента, ориентировочно ориентированный на форму приближенного поиска в массиве элементов:



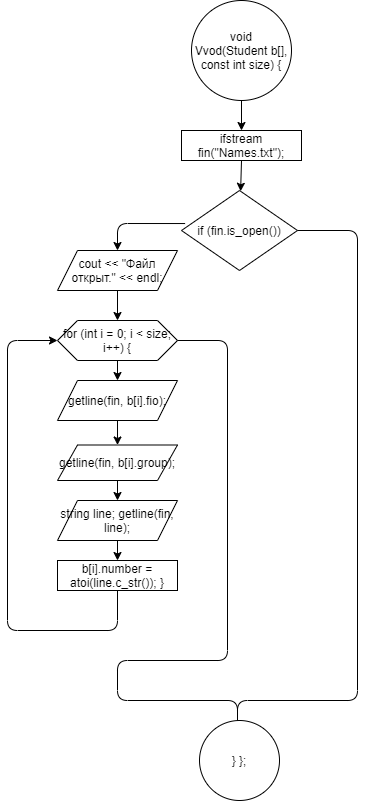
**5.4.** Линейный поиск основан на банальном переборе всех элементов структуры с постепенным сравнением всех полученных таким образом элементов с искомым :

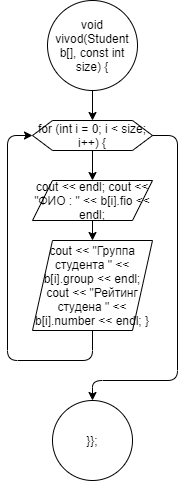


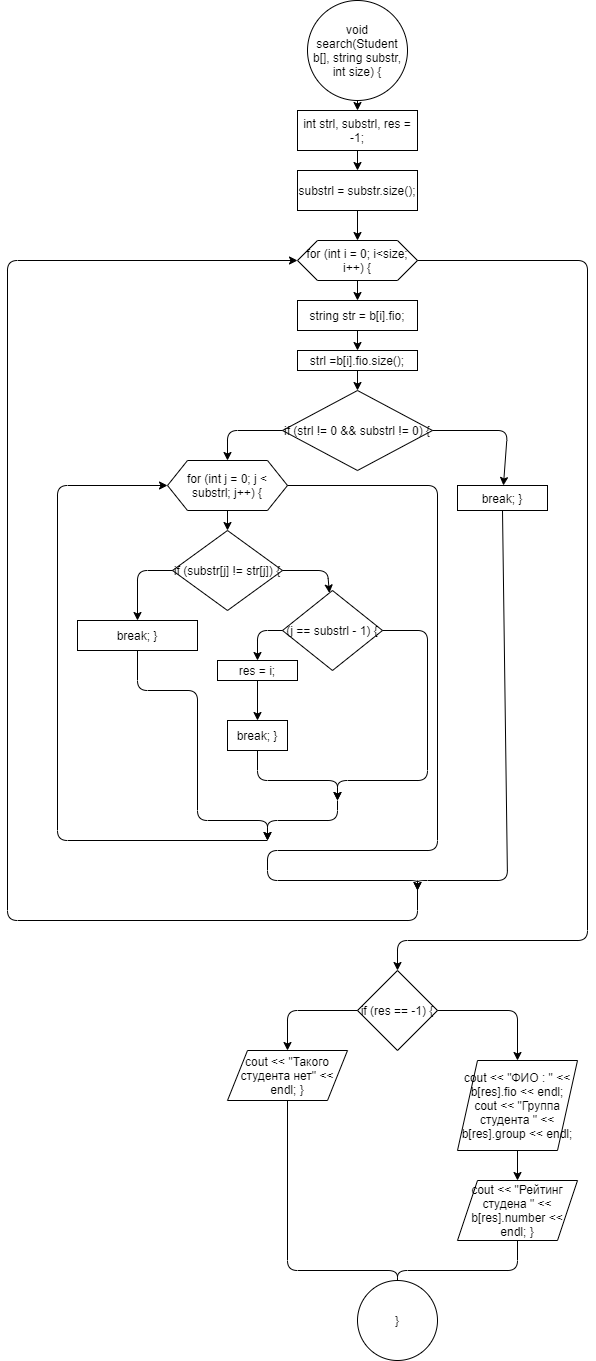
**5.4.** Организация меню основана на приеме введенного пользователем значения и её дальнейшей обработкой, завязанной на многоступенчатом условии if:

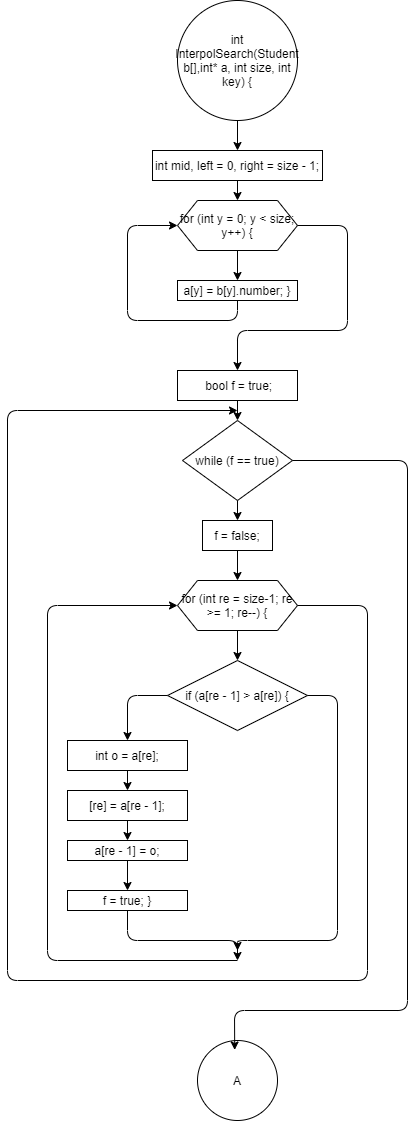


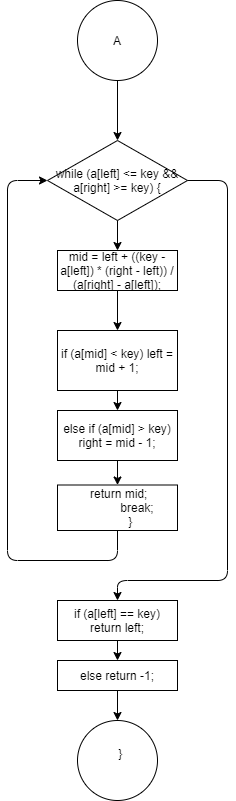
**Блок-схема программы**

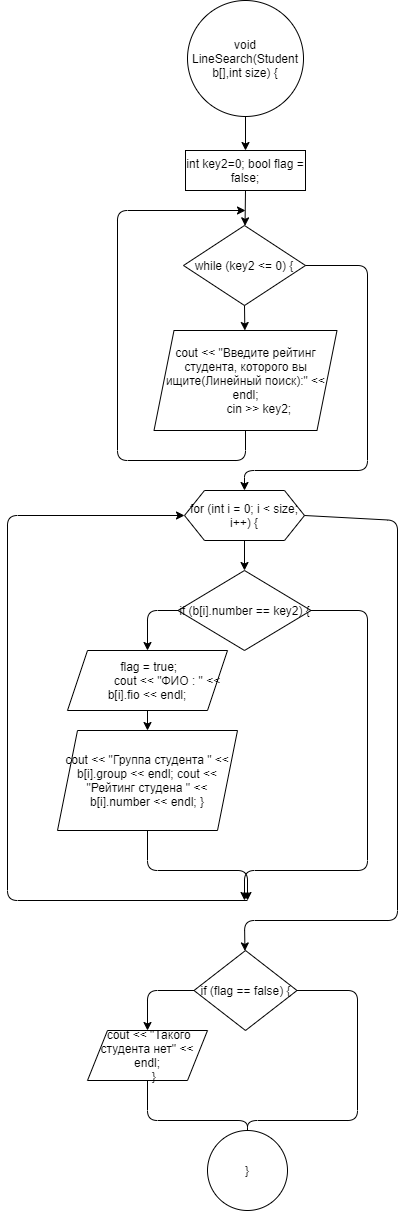


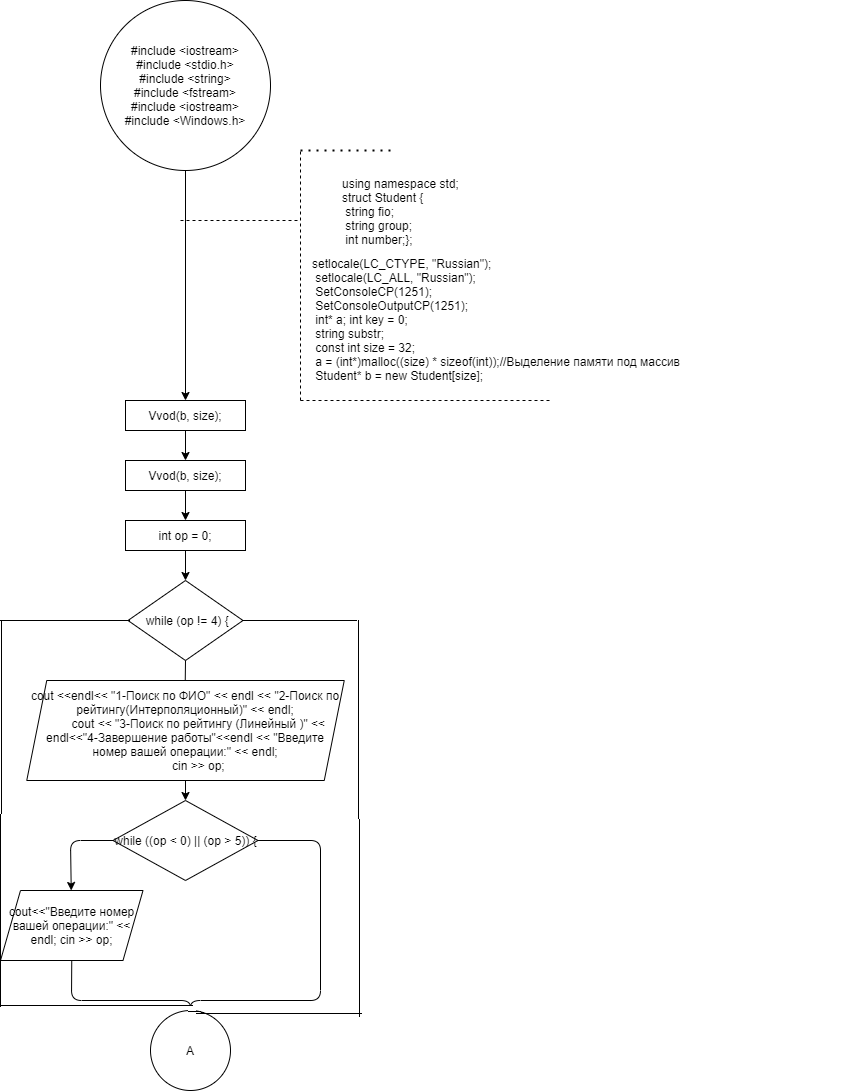


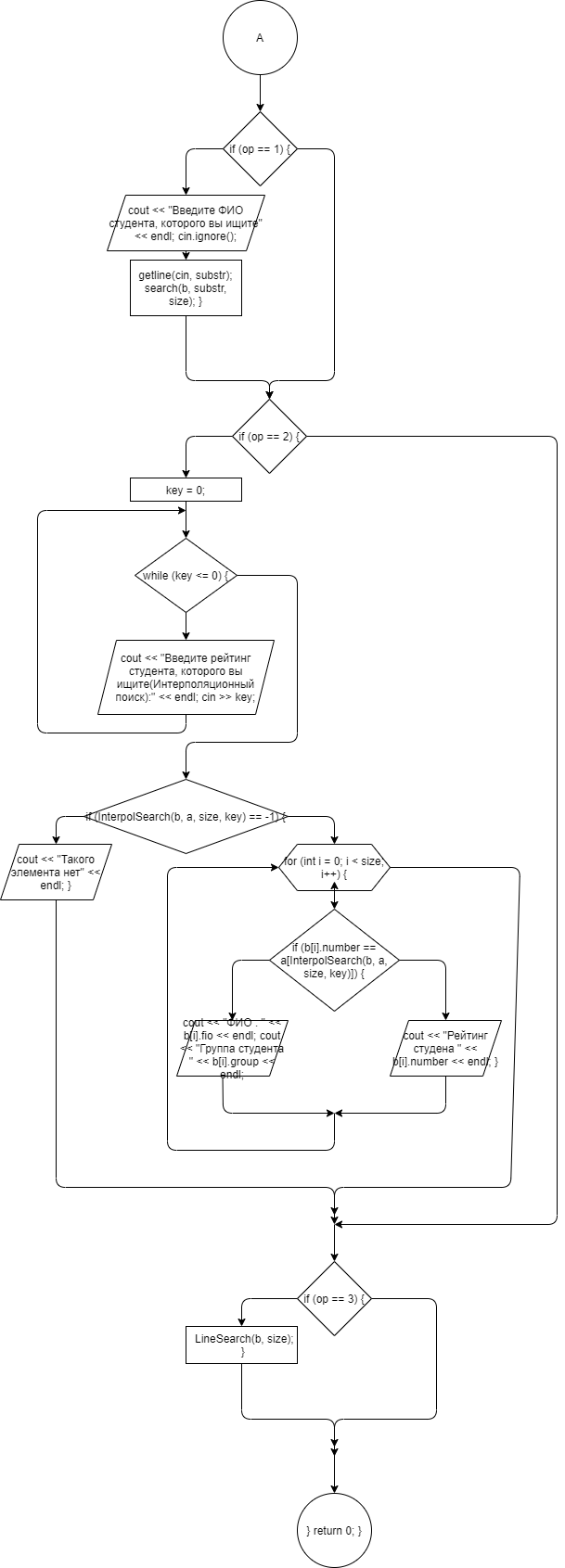












**Решение**

#include <iostream>

#include <stdio.h>

#include <string>

#include <fstream>

#include <iostream>

#include <Windows.h>

using namespace std;

struct Student {

string fio;

string group;

int number;

};

void Vvod(Student b[], const int size) {

ifstream fin("Names.txt");

if (fin.is\_open())//Если открытие файла прошло успешно

{

cout << "Файл открыт." << endl;

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

getline(fin, b[i].fio);

getline(fin, b[i].group);

string line;

getline(fin, line);

b[i].number = atoi(line.c\_str());

}

}

};

void vivod(Student b[], const int size) {

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

cout << endl;

cout << "ФИО : " << b[i].fio << endl;

cout << "Группа студента " << b[i].group << endl;

cout << "Рейтинг студена " << b[i].number << endl;

}

};

void search(Student b[], string substr, int size) {

int strl, substrl, res = -1;

substrl = substr.size();

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

string str = b[i].fio;

strl =b[i].fio.size();

if (strl != 0 && substrl != 0) {

for (int j = 0; j < substrl; j++) {

if (substr[j] != str[j]) {

break;

}

else if (j == substrl - 1) {

res = i;

break;

}

}

}

else break;

}

if (res == -1) {

cout << "Такого студента нет" << endl;

}

else {

cout << "ФИО : " << b[res].fio << endl;

cout << "Группа студента " << b[res].group << endl;

cout << "Рейтинг студена " << b[res].number << endl;

}

}

int InterpolSearch(Student b[],int\* a, int size, int key) {

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

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

a[y] = b[y].number;

}

bool f = true;

while (f == true)

{

f = false;

for (int re = size-1; re >= 1; re--)

{

if (a[re - 1] > a[re])

{

int o = a[re];

a[re] = a[re - 1];

a[re - 1] = o;

f = true;

}

}

}

while (a[left] <= key && a[right] >= key) {

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

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

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

else {

return mid;

break;

}

}

if (a[left] == key) return left;

else return -1;

}

void LineSearch(Student b[],int size) {

int key2=0;

bool flag = false;

while (key2 <= 0) {

cout << "Введите рейтинг студента, которого вы ищите(Линейный поиск):" << endl;

cin >> key2;

}

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

if (b[i].number == key2) {

flag = true;

cout << "ФИО : " << b[i].fio << endl;

cout << "Группа студента " << b[i].group << endl;

cout << "Рейтинг студена " << b[i].number << endl;

}

}

if (flag == false) {

cout << "Такого студента нет" << endl;

}

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

setlocale(LC\_ALL, "Russian");

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int\* a;

int key = 0;

string substr;

const int size = 32;

a = (int\*)malloc((size) \* sizeof(int));//Выделение памяти под массив

Student\* b = new Student[size];

Vvod(b, size);

vivod(b, size);

int op = 0;

while (op != 4) {

cout <<endl<< "1-Поиск по ФИО" << endl << "2-Поиск по рейтингу(Интерполяционный)" << endl;

cout << "3-Поиск по рейтингу (Линейный )" << endl<<"4-Завершение работы"<<endl << "Введите номер вашей операции:" << endl;

cin >> op;

while ((op < 0) || (op > 5)) {

cout<<"Введите номер вашей операции:" << endl;

cin >> op;

}

if (op == 1) {

cout << "Введите ФИО студента, которого вы ищите" << endl;

cin.ignore();

getline(cin, substr);

search(b, substr, size);

}

if (op == 2) {

key = 0;

while (key <= 0) {

cout << "Введите рейтинг студента, которого вы ищите(Интерполяционный поиск):" << endl;

cin >> key;

}

if (InterpolSearch(b, a, size, key) == -1) {

cout << "Такого элемента нет" << endl;

}

else {

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

if (b[i].number == a[InterpolSearch(b, a, size, key)]) {

cout << "ФИО : " << b[i].fio << endl;

cout << "Группа студента " << b[i].group << endl;

cout << "Рейтинг студена " << b[i].number << endl;

}

}

}

}

if (op == 3) {

LineSearch(b, size);

}

}

return 0;

}

**Скриншоты результатов работы программы**

