МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Липецкий Государственный Технический Университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

Лабораторная работа

по технологиям программирования №2

“Методы сортировки и поиска в информационных массивах”

Студент \_\_\_\_\_\_\_\_\_\_\_\_ Станиславчук С.М.

(подпись, дата)

Группа АС-21-1

Руководитель

Доцент, кандидат наук \_\_\_\_\_\_\_\_\_\_\_\_ Хромов А. М.

(подпись, дата)

Липецк 2022 г.

**Содержание:**

1. Титульный лист.

2. Задание кафедры, соответствующее варианту, номер варианта.

3. Цель работы.

4. Краткие теоретические сведения

5. Блок-схему алгоритма разработанной программы

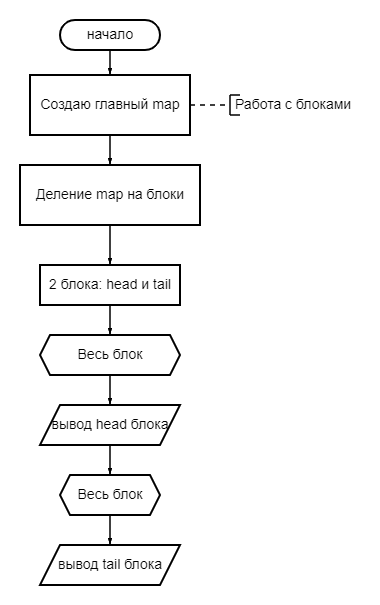
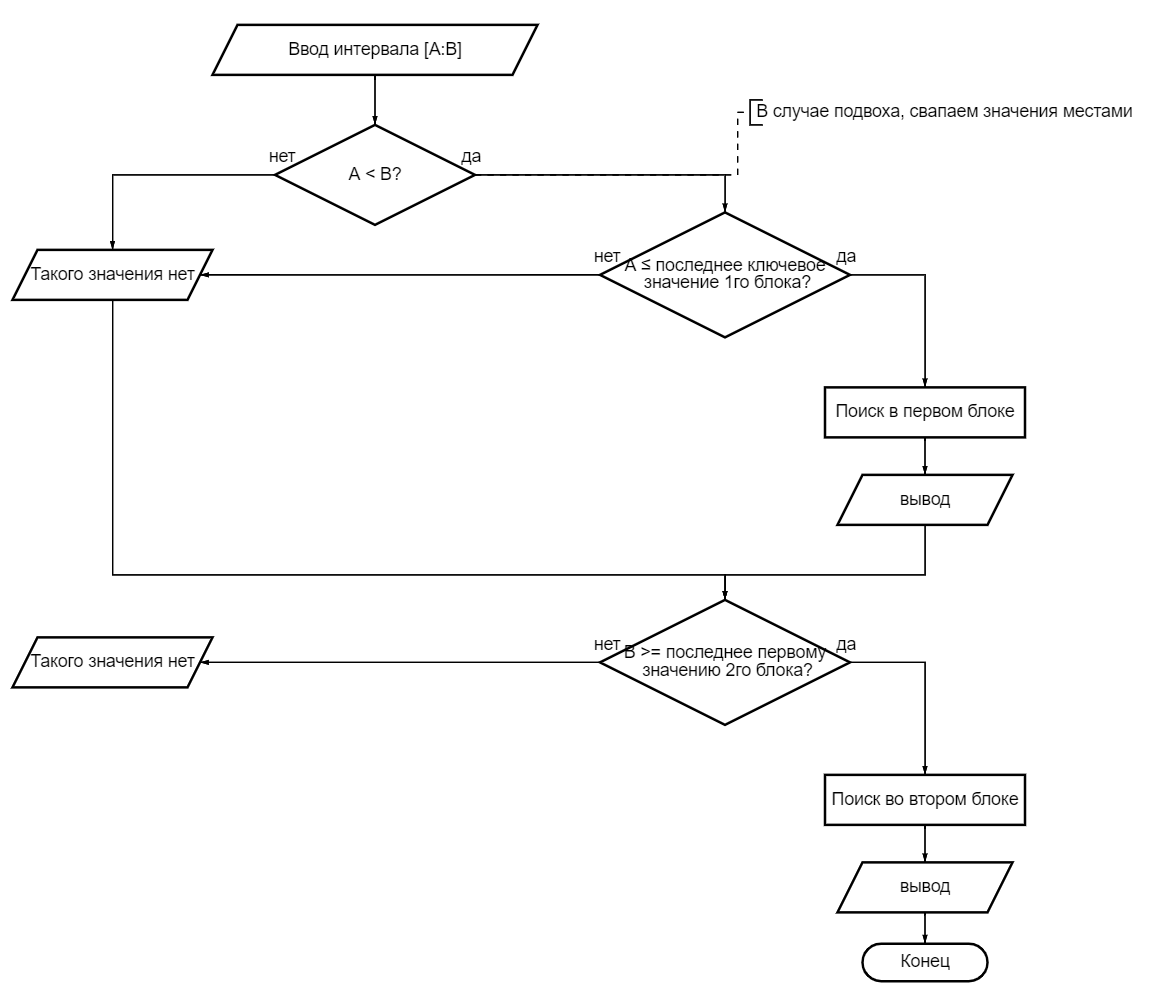
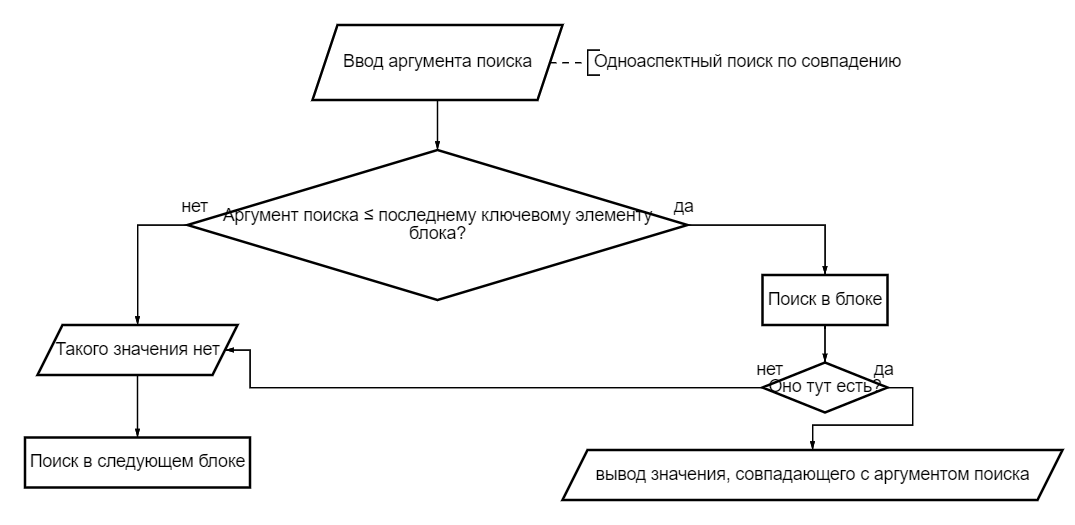
6. Текст программы

7. Контрольный пример

8. Выводы по работе.

**2. Задание кафедры, соответствующее варианту, номер варианта.**Написать программу, реализующую один из алгоритмов программного поиска данных в информационном массиве, расположенном в оперативной памяти, используя выбранные в соответствии с вариантом из таблицы формат ключа, формат других полей записи, вид и метод поиска.

**3. Цель работы:** приобретение навыков реализации базовых алгоритмов программного поиска в информационных массиве.

**4. Блок-схема алгоритма разработанной программы  
**

**5. Текст программы.**/\* Написать программу, реализующую один из алгоритмов программного

поиска данных в информационном массиве, расположенном в оперативной памяти, используя формат ключа: char; формат инфо полей записи: int[], char; тип информационного поиска: одноаспектный поиск по совпадению и одноаcпектный поиск по интервалу, метод поиска: Блочный поиск \*/

// Программа идеальна ( ͡~ ͜ʖ ͡°), почти

#include <iostream>

#include <locale.h>

#include <stdlib.h>

#include <map>

#include <string>

#include <math.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RUS");

struct info

{

int numbers[2]; //значение объёма массива должно быть const

char CHAR;

};

int i;

char name, min, max;

map <char, info> my\_map = { {'H', {1 , 5, 'h'}},

{'K', {2 , 3, 'k'}},

{'A', {2 , 4, 'a'}},

{'D', {9 , 0, 'd'}},

{'C', {5 , 6, 'c'}},

{'S', {3 , 4, 's'}},

{'F', {7 , 2, 'f'}},

{'G', {2 , 7, 'g'}},

{'Z', {4 , 4, 'z'}},

{'Q', {1 , 3, 'z'}},

{'J', {7 , 0, 'z'}},

{'B', {4 , 7, 'b'}} };

// Блоки

char name\_for\_block; int tmp = 0;

auto size = my\_map.size(); //размерность map

auto N\_blocks = size / 2; //число элементов в 1 блоке

auto it1 = my\_map.begin();

advance(it1, N\_blocks);

auto head\_block = map{

make\_move\_iterator(my\_map.begin()),

make\_move\_iterator(it1)

};

auto tail\_block = map{

make\_move\_iterator(it1),

make\_move\_iterator(my\_map.end())

};

cout << "Block 1\n"; //вывод 1-го блока

for (auto [key, value] : head\_block) {

cout << key << " => " << value.numbers[0] << " " << value.numbers[1] << " " << value.CHAR << "\n";

}

cout << "\n\nBlock 2\n"; //вывод 2-го блока

for (auto [key, value] : tail\_block) {

cout << key << " => " << value.numbers[0] << " " << value.numbers[1] << " " << value.CHAR << "\n";

}

cout << "\nВведите элемент поиска: ";

cin >> name\_for\_block;

//Одноаспектный поиск по совпадению

//ищем в 1-ом блоке

if (tmp == 0)

{

cout << "\n\nBlock 1\n";

if (name\_for\_block <= head\_block.rbegin()->first)

{

auto it = head\_block.find(name\_for\_block);

if (it != head\_block.end()) {

cout << it->first << " => " << it->second.numbers[0] << " " << it->second.numbers[1] << " " << it->second.CHAR << endl << endl;

tmp = 1;

}

else cout << "Такого ключевого значения нет" << endl;

tmp = 0;

}

}

else {

cout << "Такого ключевого значения нет" << endl;

tmp = 0;

}

//ищем во 2-ом блоке

if (tmp == 0)

{

cout << "\n\nBlock 2\n";

if (name\_for\_block <= tail\_block.rbegin()->first)

{

auto it = tail\_block.find(name\_for\_block);

if (it != tail\_block.end()) {

cout << it->first << " => " << it->second.numbers[0] << " " << it->second.numbers[1] << " " << it->second.CHAR << endl << endl;

tmp = 1;

}

else cout << "Такого ключевого значения нет" << endl;

tmp = 0;

}

}

else {

cout << "Такого ключевого значения нет" << endl;

tmp = 0;

}

// Одноаспектный поиск по интервалу

cout << endl << "Одноаспектный поиск по интервалу [";

cin >> min; system("cls");

cout << "Одноаспектный поиск по интервалу[" << min << ";";

cin >> max; system("cls");

cout << "Одноаспектный поиск по интервалу[" << min << ";" << max << "]" << endl;

//////////////

if (min < max)

{

cout << "\n\nBlock 1\n";

if (min <= head\_block.rbegin()->first)

{

auto it1 = head\_block.lower\_bound(min), it2 = head\_block.upper\_bound(max);

for (auto pos = it1; pos != it2; ++pos)

cout << pos->first << " => " << pos->second.numbers[0] << " " << pos->second.numbers[1]

<< " " << pos->second.CHAR << endl;

}

else cout << "Такого(их) ключевого(ых) значения(й) нет" << endl;

cout << "\nBlock 2\n";

if (max >= tail\_block.begin()->first)

{

auto it3 = tail\_block.lower\_bound(min), it4 = tail\_block.upper\_bound(max);

for (auto pos = it3; pos != it4; ++pos)

cout << pos->first << " => " << pos->second.numbers[0] << " " << pos->second.numbers[1]

<< " " << pos->second.CHAR << endl;

}

else cout << "Такого(их) ключевого(ых) значения(й) нет" << endl;

}

else // на случай подвоха

{

cout << "\n\nBlock 1\n";

if (min >= head\_block.begin()->first)

{

auto it1 = head\_block.upper\_bound(min), it2 = head\_block.lower\_bound(max); swap(it1, it2);

for (auto pos = it1; pos != it2; ++pos)

cout << pos->first << " => " << pos->second.numbers[0] << " " << pos->second.numbers[1]

<< " " << pos->second.CHAR << endl;

}

else cout << "Такого(их) ключевого(ых) значения(й) нет" << endl;

cout << "\nBlock 2\n";

if (max <= tail\_block.rbegin()->first)

{

auto it3 = tail\_block.upper\_bound(min), it4 = tail\_block.lower\_bound(max); swap(it3, it4);

for (auto pos = it3; pos != it4; ++pos)

cout << pos->first << " => " << pos->second.numbers[0] << " " << pos->second.numbers[1]

<< " " << pos->second.CHAR << endl;

}

else cout << "Такого(их) ключевого(ых) значения(й) нет" << endl;

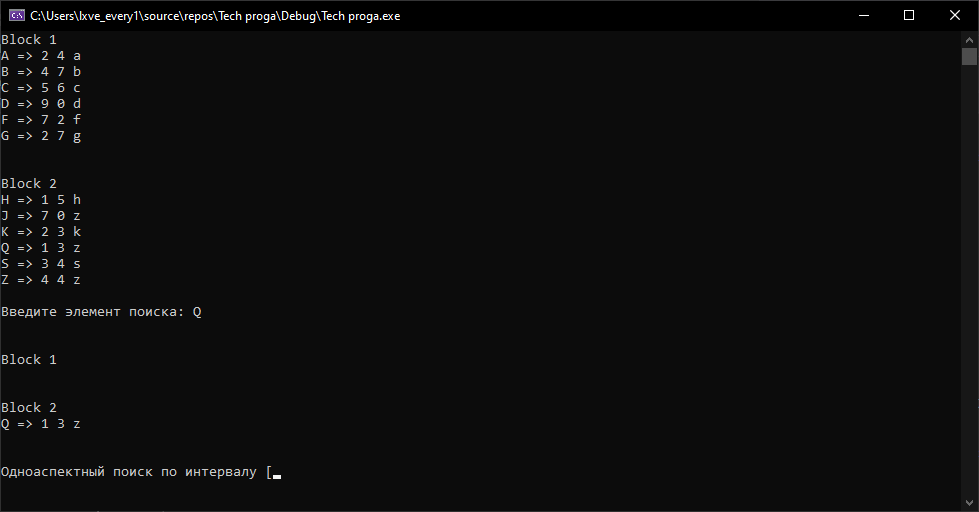
}

return 0;

}

**6. Контрольный пример**

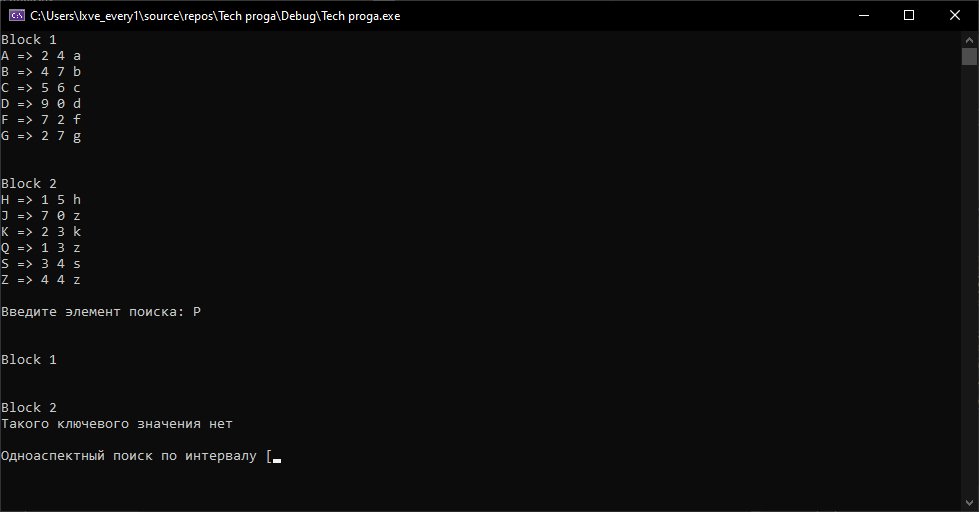
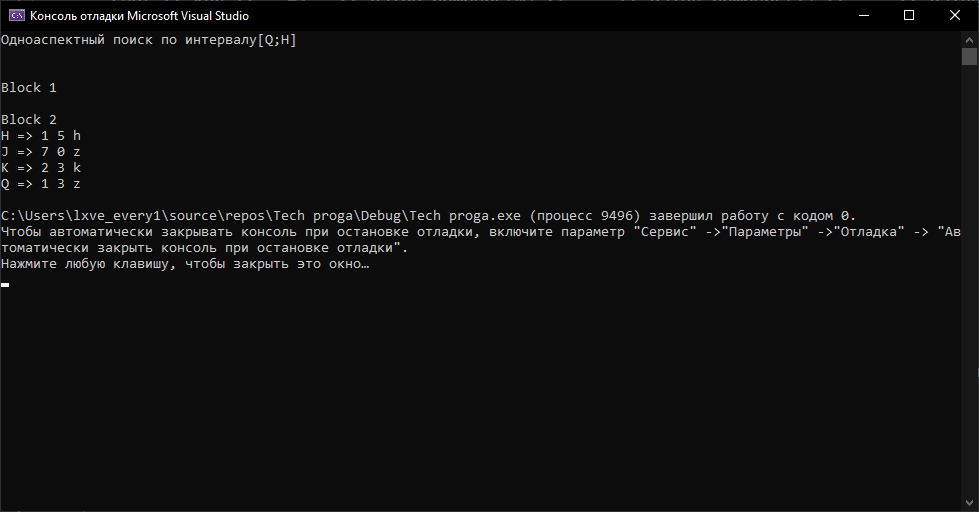
**Пример 1.**Вводим Q:



В поиске по интервалу вводим [A:S]

  
  
**Пример 2.**

Введем несуществующий элемент в памяти: P

Ввели интервал [Q:H]  


Программа успешно выполнилась.

**7. Выводы по работе.**Приобрёл навыки реализации базовых алгоритмов программного поиска в информационных массиве.