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 №12**

Дисциплина: основы алгоритмизации и программирования

Тема: “Поиск” в C++

Вариант 24

Выполнил работу

студент группы РИС-20-1б

Нечаев Д.А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Цель задачи**

Цель – написать программу для осуществления прямого поиска подстроки в строке, интерполяционного поиска и линейного поиска

Для достижения поставленной цели необходимо решить следующие задачи:

* Проанализировать алгоритмы прямого поиска подстроки в строке, интерполяционного поиска и линейного поиска
* Понять, как воспользоваться алгоритмами поиска, в условиях поставленной задачи
* Проанализировать инструменты ввода и вывода данных

Постановка задачи

Выполнить поиск элемента в массиве по ключу в соответствии с вариантом, используя методы прямого поиска подстроки в строке и интерполяционного поиска

Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

* Создание структуры данных, среди которых будет осуществлен поиск

struct dat

{

string fio;

string date;

int pass;

string pass1;

};

* Создание меню с возможностью выбора метода поиска и защитой от случайных ошибок

while (menu < 1 || menu > 3)

{

cout << "Выбери метод поиска: \n 1. Поиск подстроки в строке \n 2. Интерполяционный поиск \n 3. Прямой поиск \n Ввод: ";

cin >> menu;

}

if (menu == 1) line\_in\_line();

if (menu == 2) intersearch();

if (menu == 3) clear1();

* Реализация методов поиска в соответствующих функциях

1. Для решения задачи используются переменные:

Несколько целочисленных переменных для управления циклами, контроля размеров массивов и контроля длины строк

int key, num = -1, mid = -1, left = 0, right = 4, i, temp, item;

Переменная типа string для хранения ключа

string key;

Массив структур:

dat a[5] = { N1, N2, N3, N4, N5 };

1. Ввод данных осуществляется посредством функции

cin >> key;

Вывод данных осуществляется посредством функции

F << "Искомый элемент не найден";

1. Для решения задачи будут использованы циклы. Например, цикл сортировки элементов массива по возрастанию:

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

{

int temp = a[i];

int item = i - 1;

while (item >= 0 && a[item] > temp)

{

a[item + 1] = a[item];

a[item] = temp;

item--;

}

}

1. В программе использована защита от случайных ошибок в меню

while (menu < 1 || menu > 3)

{

cout << "Выбери метод поиска: \n 1. Поиск подстроки в строке \n 2. Интерполяционный поиск \n 3. Прямой поиск \n Ввод: ";

cin >> menu;

}

1. В программе используется вывод искомого элемента и отчета в файл

F << " " << num << ") " << '\t' << a[num - 1].fio << endl << '\t' << a[num - 1].date << endl << '\t' << a[num - 1].pass << endl << "\n";

Код

#include <iostream>

#include <string>

#include <fstream>

using namespace std;

struct dat

{

string fio;

string date;

int pass;

string pass1;

};

void line\_in\_line()

{

string key;

int num = -1;

ofstream F("F.txt");

dat N1 = { "Иванов Иван Иванович", "04.10.2002", 507263, "507263" };

dat N2 = { "Петров Петр Петрович", "18.02.2010", 263598, "263598" };

dat N3 = { "Алексеев Алексей Алексеевич", "06.12.2000", 648531, "648531" };

dat N4 = { "Александров Александр Александрович", "06.10.2002", 984563, "984563" };

dat N5 = { "Максимов Максим Максимович", "30.10.1998", 651234, "651234" };

dat a[5] = { N1, N2, N3, N4, N5 };

for (int i = 0; i < 5; i++)

cout << " " << i + 1 << ") " << '\t' << a[i].fio << endl << '\t' << a[i].date << endl << '\t' << a[i].pass << endl << "\n";

cout << "Введи ключ (номер паспорта), по которому будет осуществлен поиск: "; cin >> key;

string a1[5]{ N1.pass1, N2.pass1, N3.pass1, N4.pass1, N5.pass1 };

for (int i = 0; i < 5; i++)

if (key == a1[i]) num = i + 1;

if (num != -1)

F << " " << num << ") " << '\t' << a[num-1].fio << endl << '\t' << a[num-1].date << endl << '\t' << a[num-1].pass << endl << "\n";

else

F << "Искомый элемент не найден";

}

void clear1()

{

string key;

int num = -1;

ofstream F("F.txt");

dat N1 = { "Иванов Иван Иванович", "04.10.2002", 507263, "507263" };

dat N2 = { "Петров Петр Петрович", "18.02.2010", 263598, "263598" };

dat N3 = { "Алексеев Алексей Алексеевич", "06.12.2000", 648531, "648531" };

dat N4 = { "Александров Александр Александрович", "06.10.2002", 984563, "984563" };

dat N5 = { "Максимов Максим Максимович", "30.10.1998", 651234, "651234" };

dat a[5] = { N1, N2, N3, N4, N5 };

for (int i = 0; i < 5; i++)

cout << " " << i + 1 << ") " << '\t' << a[i].fio << endl << '\t' << a[i].date << endl << '\t' << a[i].pass << endl << "\n";

cout << "Введи ключ (номер паспорта), по которому будет осуществлен поиск: "; cin >> key;

string a1[5]{ N1.pass1, N2.pass1, N3.pass1, N4.pass1, N5.pass1 };

for (int i = 0; i < 5; i++)

if (key == a1[i]) num = i + 1;

if (num != -1)

F << " " << num << ") " << '\t' << a[num - 1].fio << endl << '\t' << a[num - 1].date << endl << '\t' << a[num - 1].pass << endl << "\n";

else

F << "Искомый элемент не найден";

}

void intersearch()

{

int key, num = -1, mid = -1, left = 0, right = 4, i, temp, item;

ofstream F("F.txt");

dat N1 = { "Иванов Иван Иванович", "04.10.2002", 507263, "507263" };

dat N2 = { "Петров Петр Петрович", "18.02.2010", 263598, "263598" };

dat N3 = { "Алексеев Алексей Алексеевич", "06.12.2000", 648531, "648531" };

dat N4 = { "Александров Александр Александрович", "06.10.2002", 984563, "984563" };

dat N5 = { "Максимов Максим Максимович", "30.10.1998", 651234, "651234" };

dat a[5] = { N1, N2, N3, N4, N5 };

for (i = 0; i < 5; i++)

cout << " " << i + 1 << ") " << '\t' << a[i].fio << endl << '\t' << a[i].date << endl << '\t' << a[i].pass << endl << "\n";

cout << "Введи ключ (номер паспорта), по которому будет осуществлен поиск: "; cin >> key;

int a1[5]{ N1.pass, N2.pass, N3.pass, N4.pass, N5.pass };

for (i = 1; i < 5; i++) //Сортируем

{

temp = a1[i];

item = i - 1;

while (item >= 0 && a1[item] > temp)

{

a1[item + 1] = a1[item];

a1[item] = temp;

item--;

}

}

while (a1[left] <= key && a1[right] >= key && a1[mid] != key)

{

mid = left + ((key - a1[left]) \* (right - left)) / (a1[right] - a1[left]);

if (a1[mid] < key)

left = mid + 1;

else

if (a1[mid] > key)

right = mid - 1;

}

if (a1[mid] == key)

F << "Искомый элемент найден";

else F << "Искомый элемент не найден";

}

int main()

{

setlocale(LC\_ALL, "Russian");

system("chcp 1251>nul");

int menu = -1;

while (menu < 1 || menu > 3)

{

cout << "Выбери метод поиска: \n 1. Поиск подстроки в строке \n 2. Интерполяционный поиск \n 3. Прямой поиск \n Ввод: ";

cin >> menu;

}

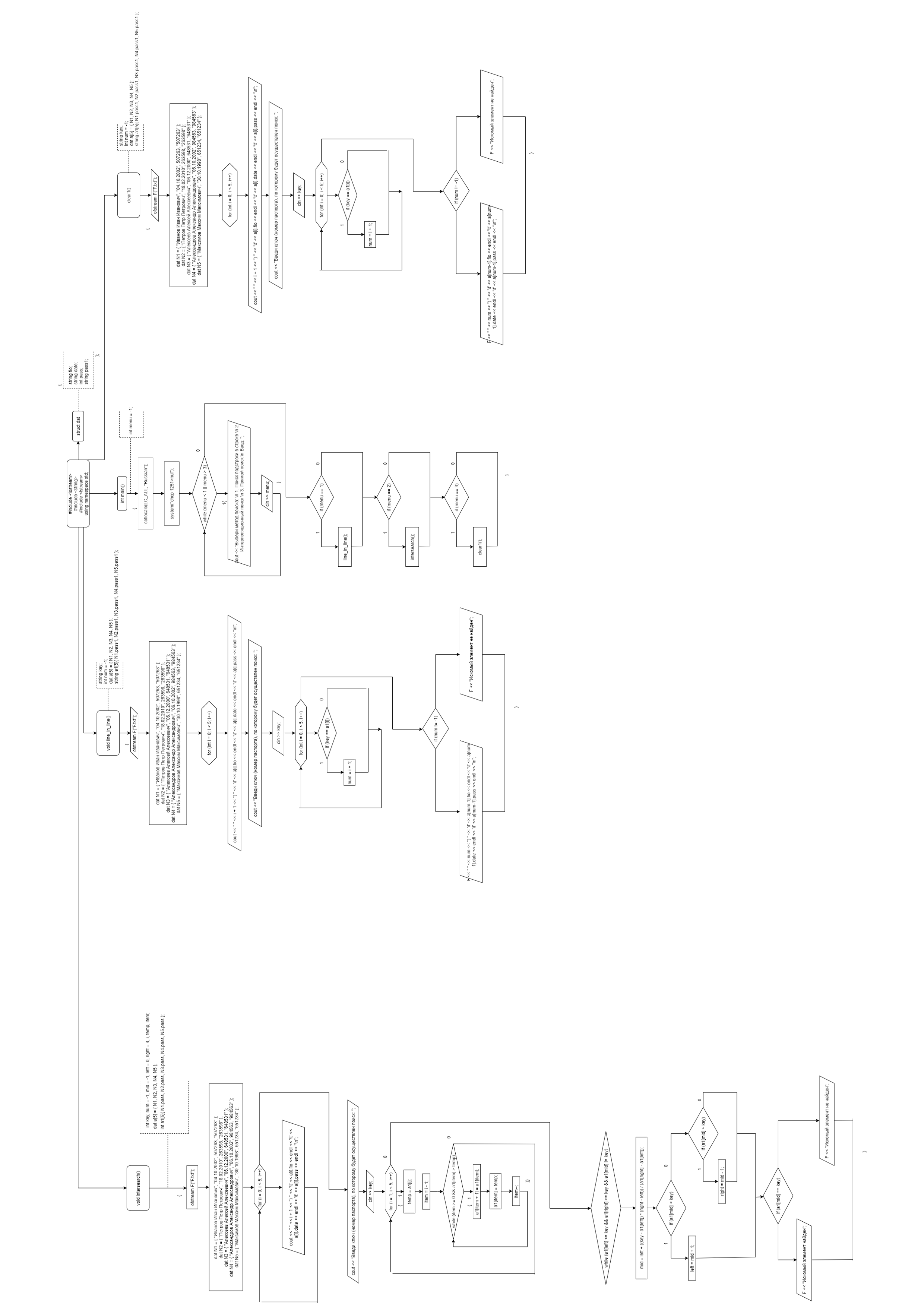
if (menu == 1) line\_in\_line();

if (menu == 2) intersearch();

if (menu == 3) clear1();

}

Блок-схема:



Работа кода

