Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра **«**Информационные технологии и автоматизированные системы**»**

направление подготовки: 09.03.04 – **«**Программная инженерия»

**Лабораторная работа**

**“Методы поиска”**

Выполнил студент гр. РИС-24-2б

Бартов Игорь Сергеевич

Проверил:

Доц. каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

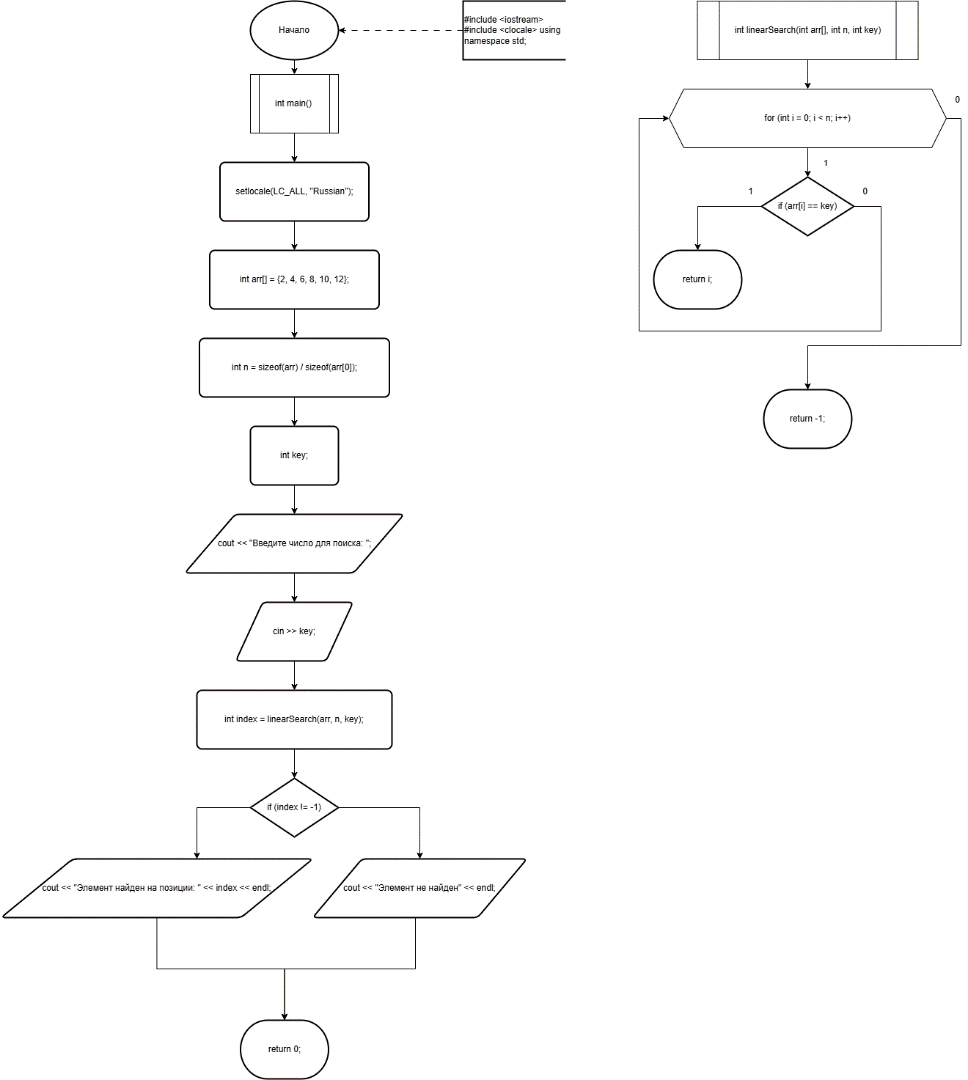
г. Пермь, 2025

1. **Постановка задачи:**

Использовать все методы поиска для нахождения позиции элемента, который равен ключу. Если элемент не найден, вернуть соответствующее сообщение.

Линейный метод

Блок-схема:



Код:

#include <iostream>

#include <clocale>

using namespace std;

int linearSearch(int arr[], int n, int key) {

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

if (arr[i] == key)

return i;

}

return -1;

}

int main() {

setlocale(LC\_ALL, "Russian");

int arr[] = { 2, 4, 6, 8, 10, 12 };

int n = sizeof(arr) / sizeof(arr[0]);

int key;

cout << "Введите число для поиска: ";

cin >> key;

int index = linearSearch(arr, n, key);

if (index != -1)

cout << "Элемент найден на позиции: " << index << endl;

else

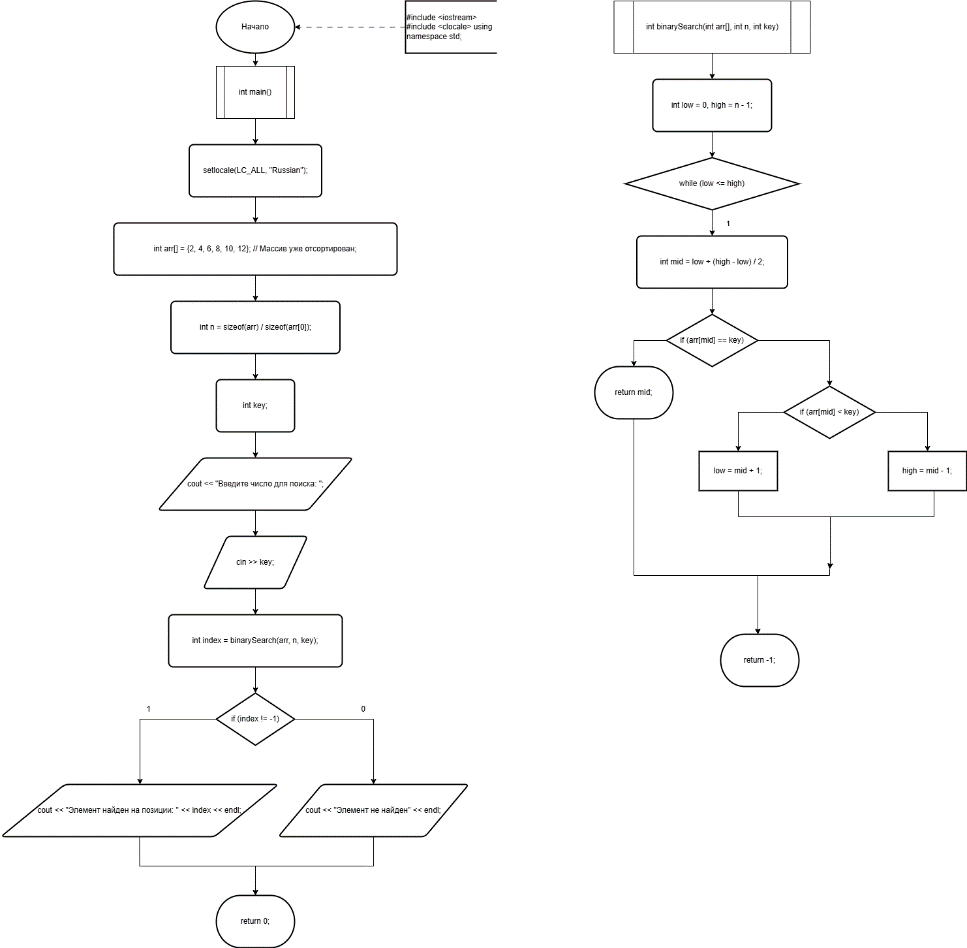
cout << "Элемент не найден" << endl;

return 0;

}

Бинарный метод

Блок-схема:



Код:

#include <iostream>

#include <clocale>

using namespace std;

int binarySearch(int arr[], int n, int key) {

int low = 0, high = n - 1;

while (low <= high) {

int mid = low + (high - low) / 2;

if (arr[mid] == key)

return mid;

else if (arr[mid] < key)

low = mid + 1;

else

high = mid - 1;

}

return -1;

}

int main() {

setlocale(LC\_ALL, "Russian");

int arr[] = { 2, 4, 6, 8, 10, 12 }; // Массив уже отсортирован

int n = sizeof(arr) / sizeof(arr[0]);

int key;

cout << "Введите число для поиска: ";

cin >> key;

int index = binarySearch(arr, n, key);

if (index != -1)

cout << "Элемент найден на позиции: " << index << endl;

else

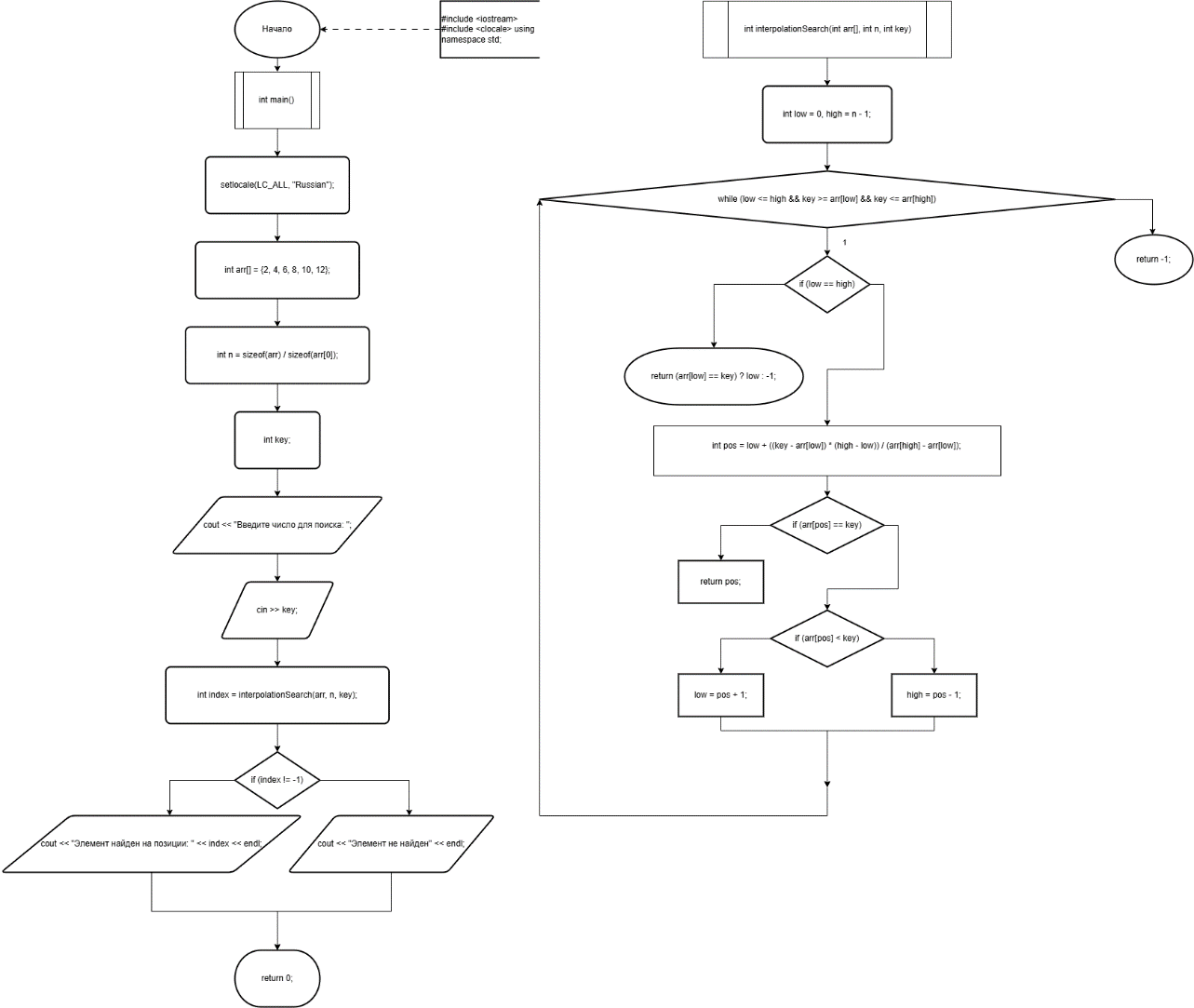
cout << "Элемент не найден" << endl;

return 0;

}

Интерполяционный метод:

Блок-схема:



Код:

#include <iostream>

#include <clocale>

using namespace std;

int interpolationSearch(int arr[], int n, int key) {

int low = 0, high = n - 1;

while (low <= high && key >= arr[low] && key <= arr[high]) {

if (low == high) {

return (arr[low] == key) ? low : -1;

}

int pos = low + ((key - arr[low]) \* (high - low)) / (arr[high] - arr[low]);

if (arr[pos] == key)

return pos;

if (arr[pos] < key)

low = pos + 1;

else

high = pos - 1;

}

return -1;

}

int main() {

setlocale(LC\_ALL, "Russian");

int arr[] = { 2, 4, 6, 8, 10, 12 };

int n = sizeof(arr) / sizeof(arr[0]);

int key;

cout << "Введите число для поиска: ";

cin >> key;

int index = interpolationSearch(arr, n, key);

if (index != -1)

cout << "Элемент найден на позиции: " << index << endl;

else

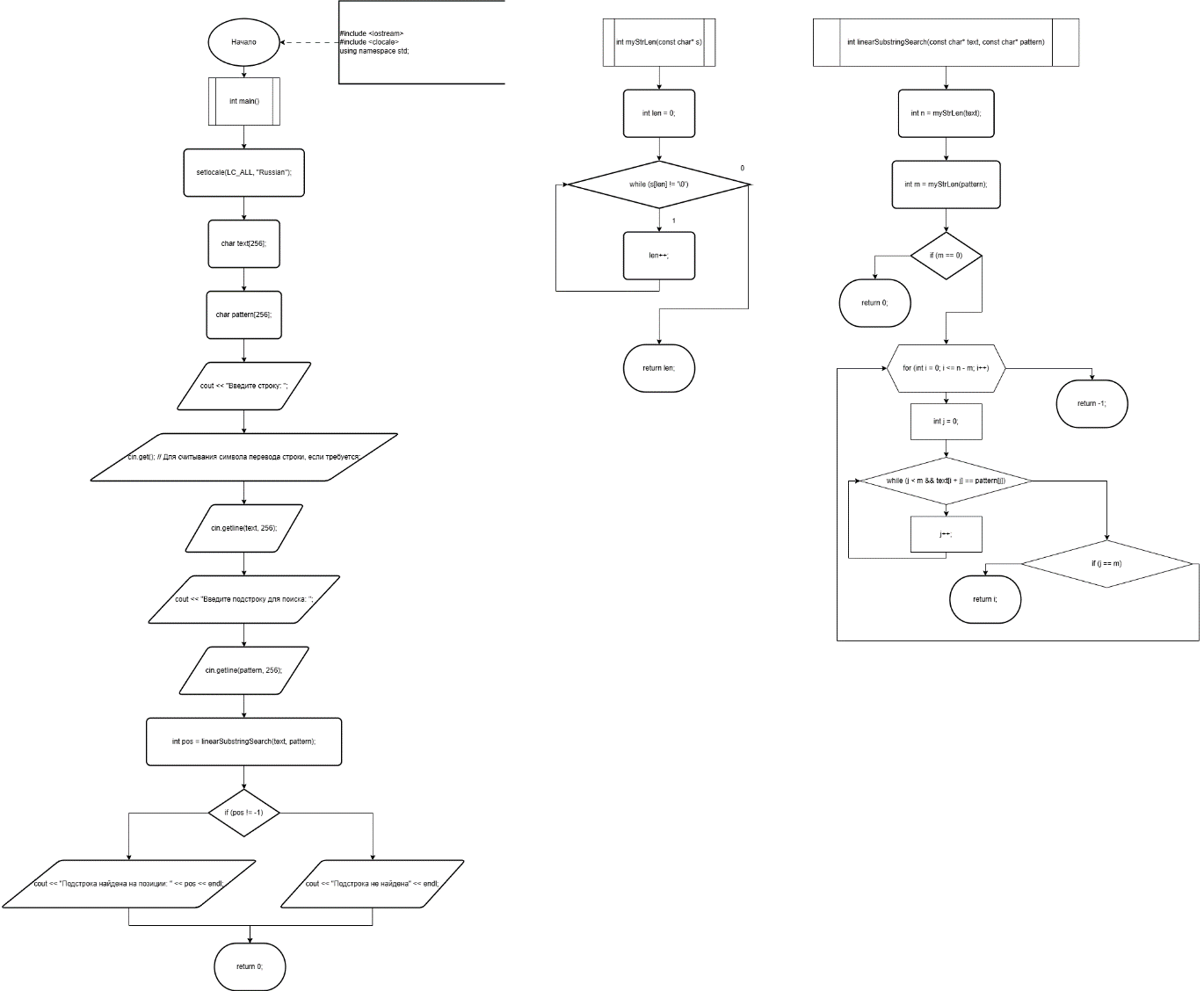
cout << "Элемент не найден" << endl;

return 0;

}

Поиск подстроки в строке:

Блок-схема:



Код:

#include <iostream>

#include <clocale>

using namespace std;

int myStrLen(const char\* s) {

int len = 0;

while (s[len] != '\0') {

len++;

}

return len;

}

int linearSubstringSearch(const char\* text, const char\* pattern) {

int n = myStrLen(text);

int m = myStrLen(pattern);

if (m == 0)

return 0;

for (int i = 0; i <= n - m; i++) {

int j = 0;

while (j < m && text[i + j] == pattern[j])

j++;

if (j == m)

return i;

}

return -1;

}

int main() {

setlocale(LC\_ALL, "Russian");

char text[256];

char pattern[256];

cout << "Введите строку: ";

cin.get();

cin.getline(text, 256);

cout << "Введите подстроку для поиска: ";

cin.getline(pattern, 256);

int pos = linearSubstringSearch(text, pattern);

if (pos != -1)

cout << "Подстрока найдена на позиции: " << pos << endl;

else

cout << "Подстрока не найдена" << endl;

return 0;

}