МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа:**

"Простые поиски"

Выполнила: студентка гр.РИС-23-3Б

Мокроусова Ангелина Андреевна

Проверила: доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2024

**Постановка задачи:**

Написать код, реализующий следущие методы поиска: линейный, бинарный, интерполяционный.

**Анализ задачи:**

1. Линейный поиск:

- Функция line принимает массив, его размер и ключ для поиска.

- Функция проходит по массиву и сравнивает каждый элемент с ключом.

- Если элемент равен ключу, функция возвращает индекс этого элемента.

- Если элемент не найден, функция возвращает -2.

- В основной функции main программа запрашивает размер массива, заполняет его случайными числами, запрашивает ключ и выводит индекс элемента или сообщение о том, что элемент не найден.

2. Бинарный поиск:

- Функция binary принимает массив, его размер и ключ для поиска.

- Функция выполняет бинарный поиск, разделяя массив пополам и сравнивая ключ с элементом в середине.

- Она продолжает деление массива до тех пор, пока не найдет ключ или не переберет весь массив.

- Возвращает индекс найденного элемента.

- В основной функции main программа запрашивает размер массива, запрашивает начальное значение первого элемента, заполняет массив и запрашивает ключ для поиска.

- После выполнения функции выводится найденный элемент или сообщение о том, что ключ не был найден.

3. Интерполяционный поиск:

- Функция interpolic принимает массив, его размер и ключ для поиска.

- Функция использует формулу интерполяции для нахождения приблизительного положения ключа в массиве.

- После приблизительного нахождения, функция сравнивает ключ с соответствующим элементом и двигается по массиву до нахождения ключа.

- Возвращает индекс найденного элемента.

- В основной функции main программа запрашивает размер массива, запрашивает начальное значение первого элемента, заполняет массив и запрашивает ключ для поиска.

- После выполнения функции выводится найденный элемент.

**Код на С++ с комментариями:**

**Код на С++ с комментариями(Линейный):**

#include<iostream>

#include<ctime>

using namespace std;

int line(int mas[], int size, int key) {

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

if (mas[i] == key)

return i;

}

return -2;

}

int main() {

int n, key;

setlocale(LC\_ALL, "ru");

srand(time(0));

cout << "Введите размер массива: ";

cin >> n;

int\* mas = new int[n];

cout << "Cлучайно сгенерированный массив: ";

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

mas[i] = rand() % 100;

cout << mas[i] << " ";

}

cout << endl << "Введите необходимый элемент(ключ): ";

cin >> key;

cout << endl << "Необходимый элемент в массиве находится под номером: " << line(mas, n, key) + 1;

}

**Код на С++ с комментариями(Бинарный):**

#include<iostream>

#include<ctime>

using namespace std;

int binarnai(int mas[], int size, int key) {

bool flag = false;

int left = 0;

int right = size;

int pivot;

while ((left <= right) && (flag != 1)) {

pivot = (left + right) / 2;

if (mas[pivot] == key) {

flag = true;

}

if (mas[pivot] > key) {

right = pivot - 1;

}

else

{

left = pivot + 1;

}

}

return pivot;

}

int main() {

int n, k, key;

setlocale(LC\_ALL, "ru");

srand(time(0));

cout << "введите размер массива: ";

cin >> n;

cout << "ввдите значение первого элемента массива: ";

cin >> k;

int\* mas = new int[n];

cout << "cгенерированный упорядоченый массив: ";

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

mas[i] = i + k;

cout << mas[i] << " ";

}

cout << endl << "введите необходимый элемент(ключ): ";

cin >> key;

int p = binarnai(mas, n, key) + 1;

if (p <1 || p>n) {

cout << "ключ не найден" ;

return 0;

}

cout << endl << "необходимый элемент в массиве находится под номером: " << p;

}

**Код на С++ с комментариями(Интерполяционный):**

#include<iostream>

#include<ctime>

using namespace std;

int interpolic(int mas[], int size, int key) {

int left = 0;

int right = size;

int pivot = -1;

bool flag = 0;

while ((left <= right) && flag != 1) {

pivot = left + ((key - mas[left] \* (right - left)) / (mas[right] - mas[left]));

if (mas[pivot] < key) left = pivot + 1;

else if (mas[pivot] > key) right = pivot + 1;

else flag = 1;

}

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

else if (mas[right] == key) return right;

else return -2;

}

int main() {

int n, k, key;

setlocale(LC\_ALL, "ru");

srand(time(0));

cout << "введите размер массива: ";

cin >> n;

cout << "ввдите значение первого элемента массива: ";

cin >> k;

int\* mas = new int[n];

cout << "cгенерированный упорядоченый массив: ";

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

mas[i] = i + k;

cout << mas[i] << " ";

}

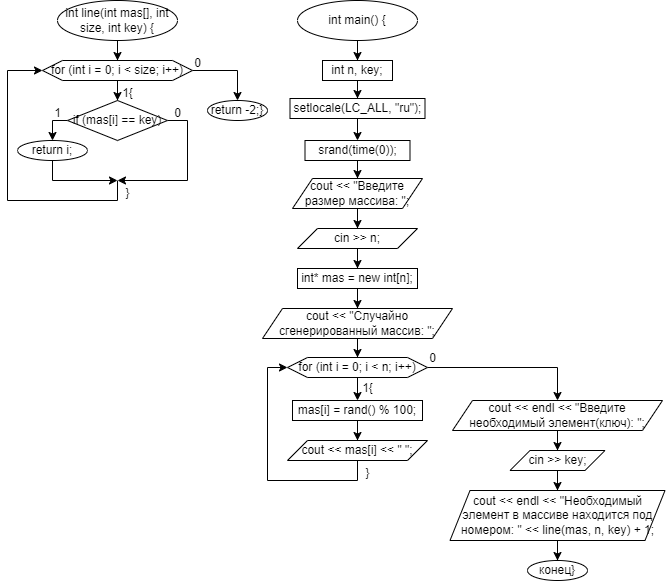
cout << endl << "введите необходимый элемент(ключ): ";

cin >> key;

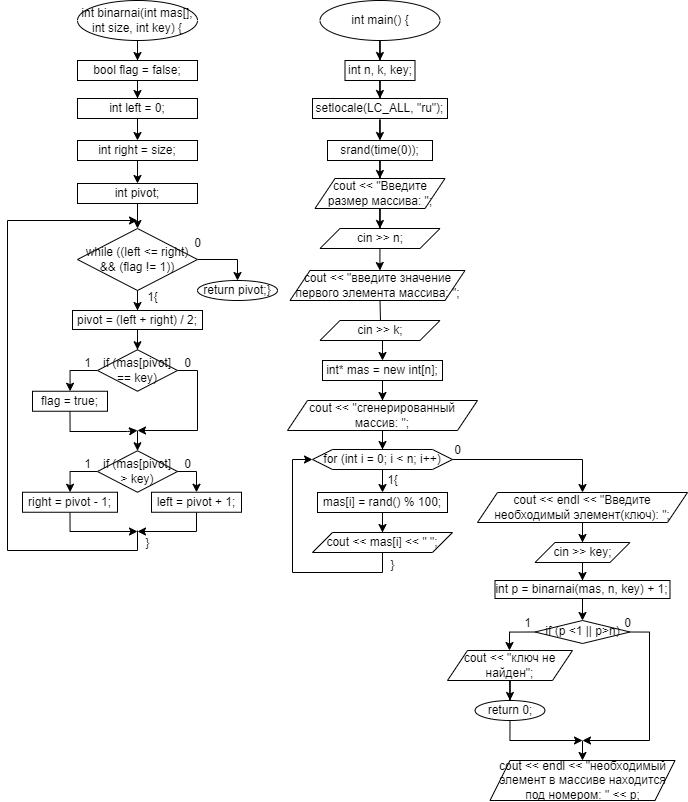
cout << endl << "необходимый элемент в массиве находится под номером: " << interpolic(mas, n, key) + 1;

}

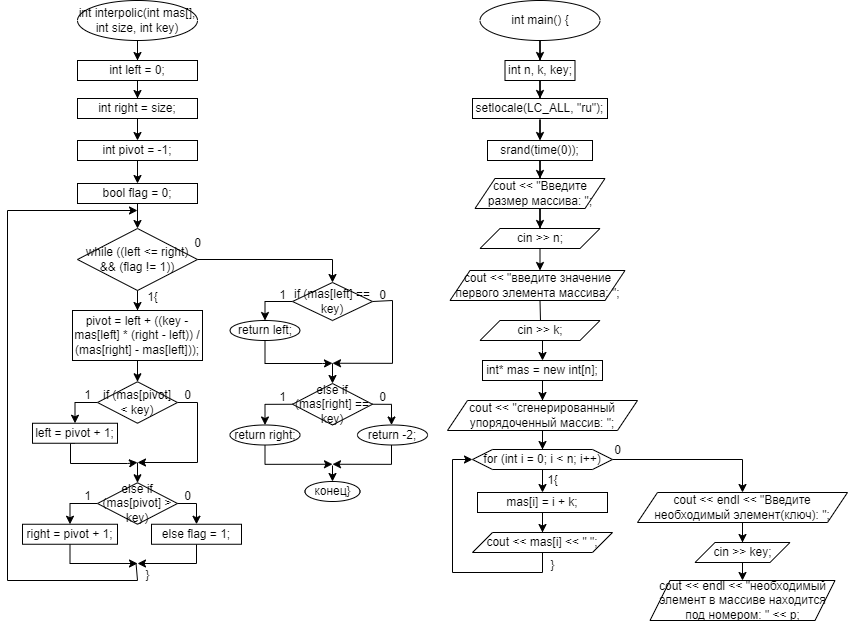
**Блок схема:**

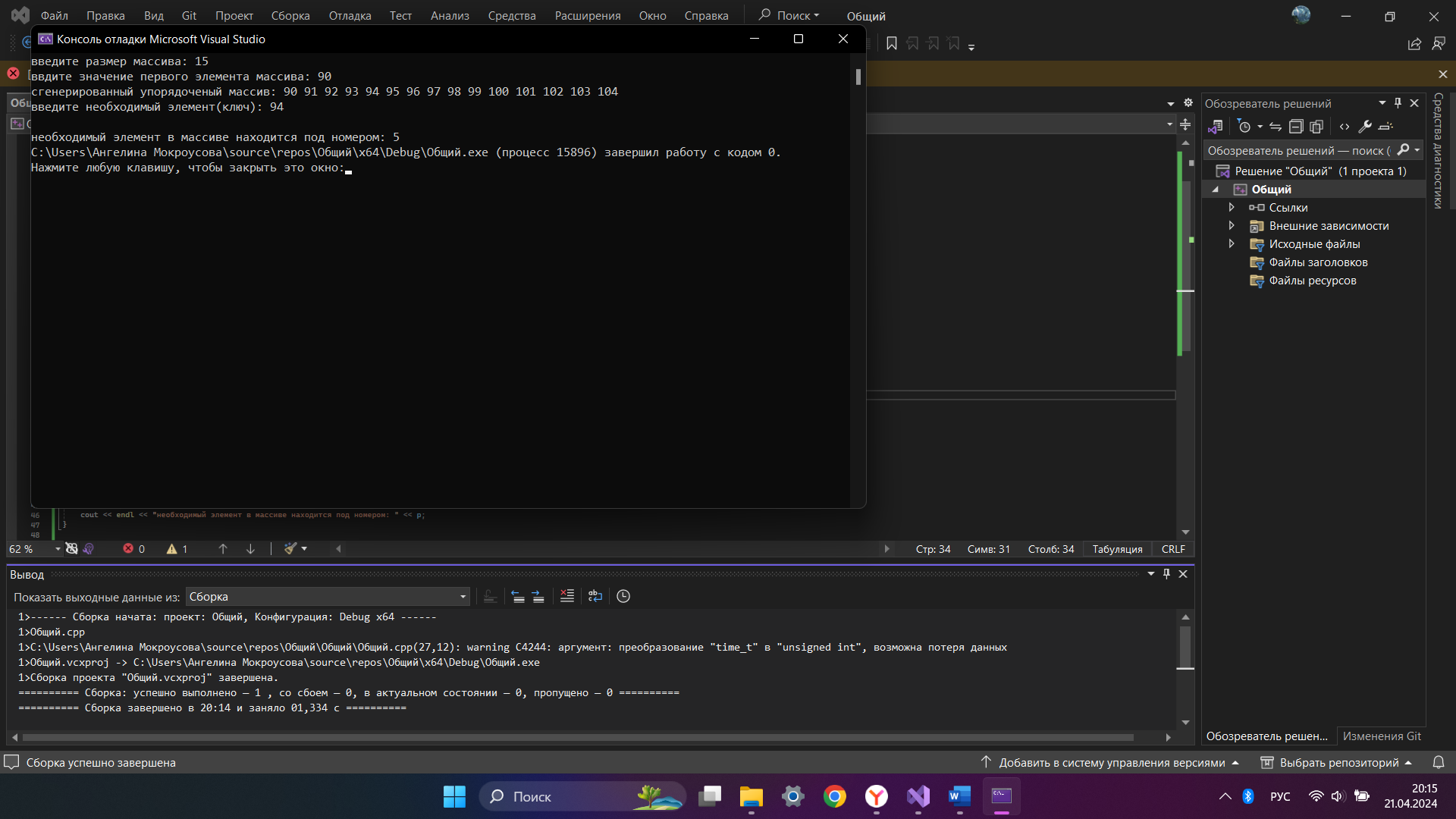
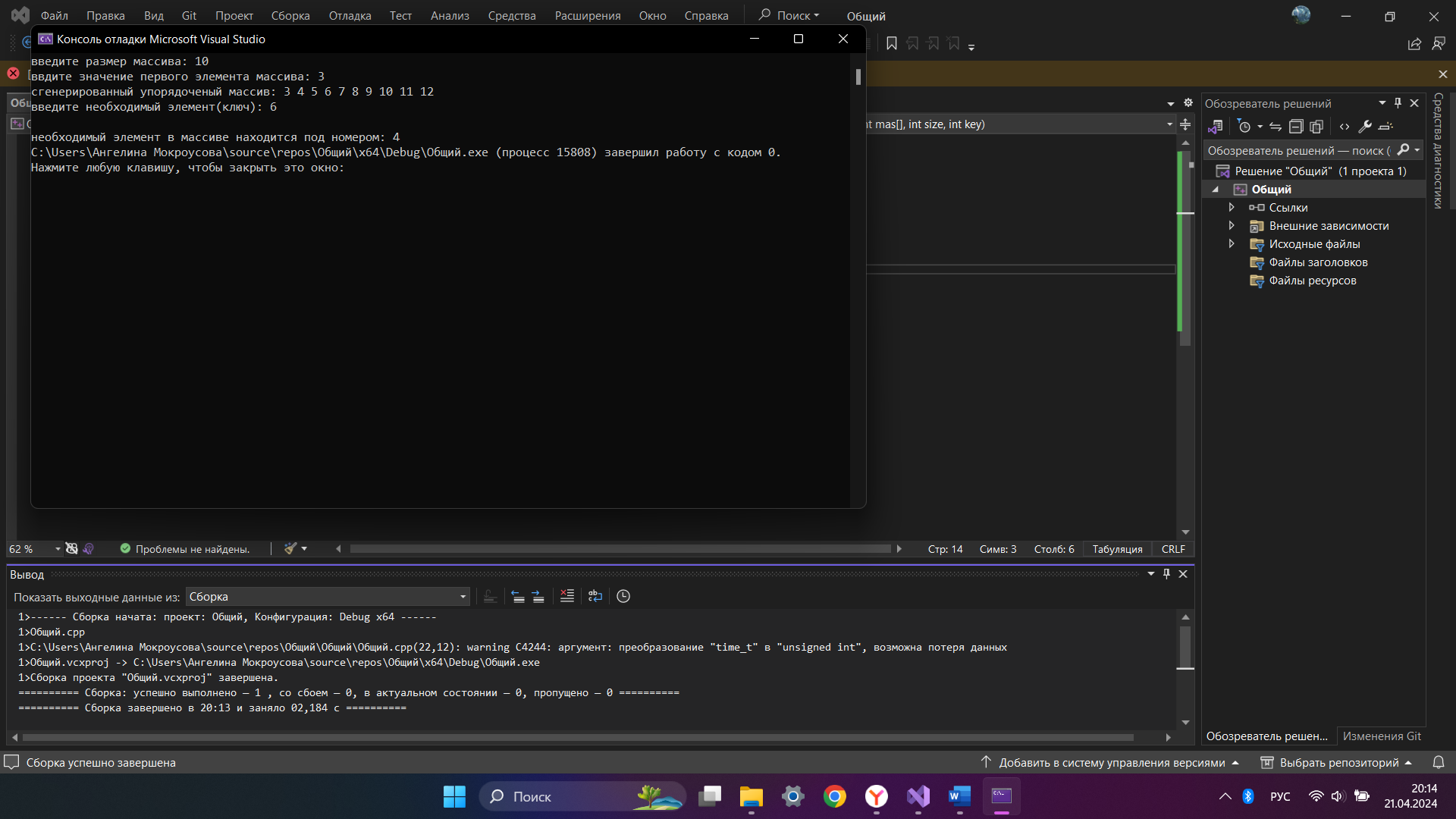
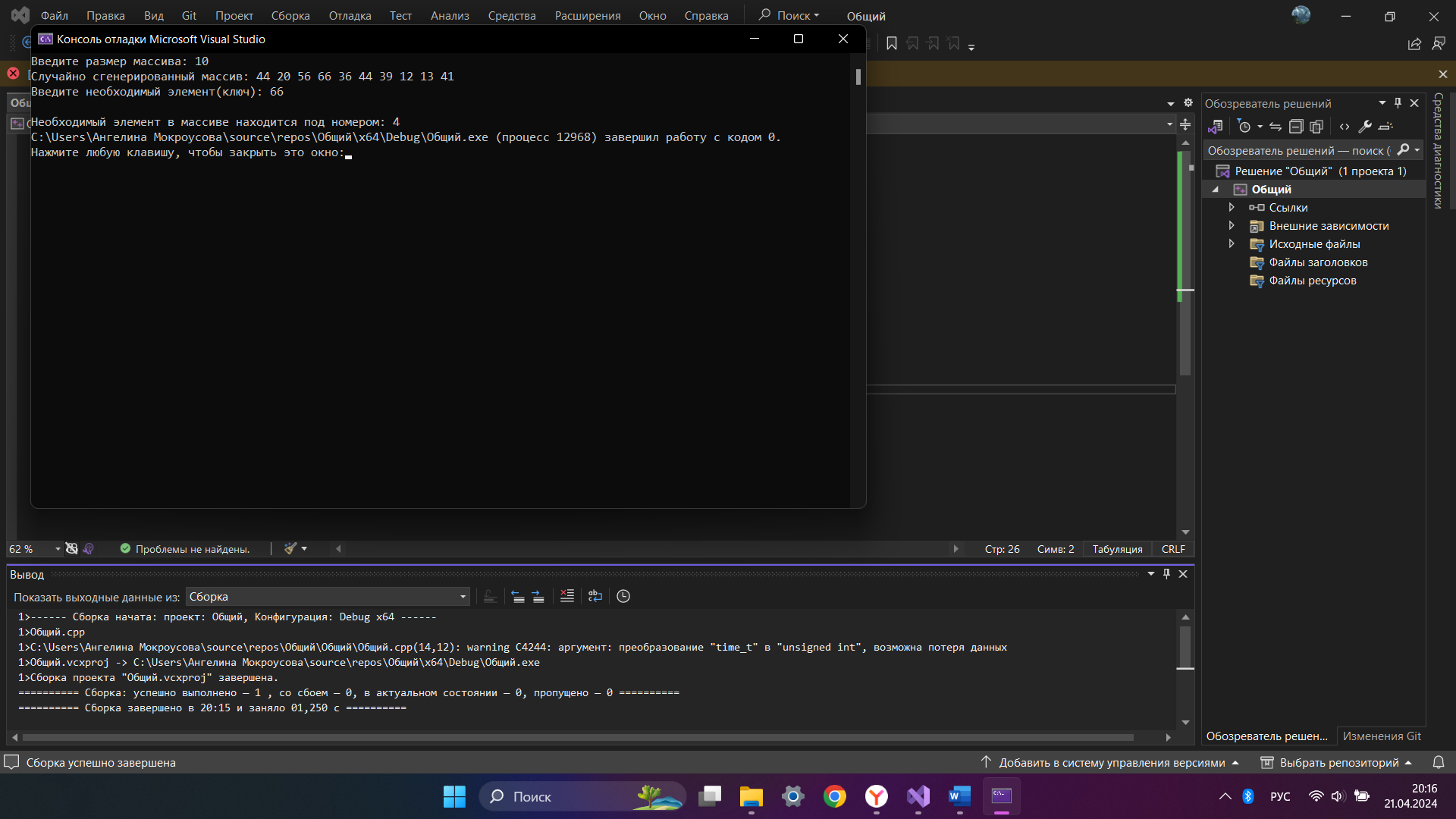
**Линейный поиск:**

**Бинарный поиск:**

****

**Интерполяционный поиск:**

**Скриншот работы программы:**



**Вывод:**

Программа успешно выполняет поставленную задачу.

**Скриншот с GitHab:**

