ЗВІТ

про виконання лабораторної роботи № 6.1

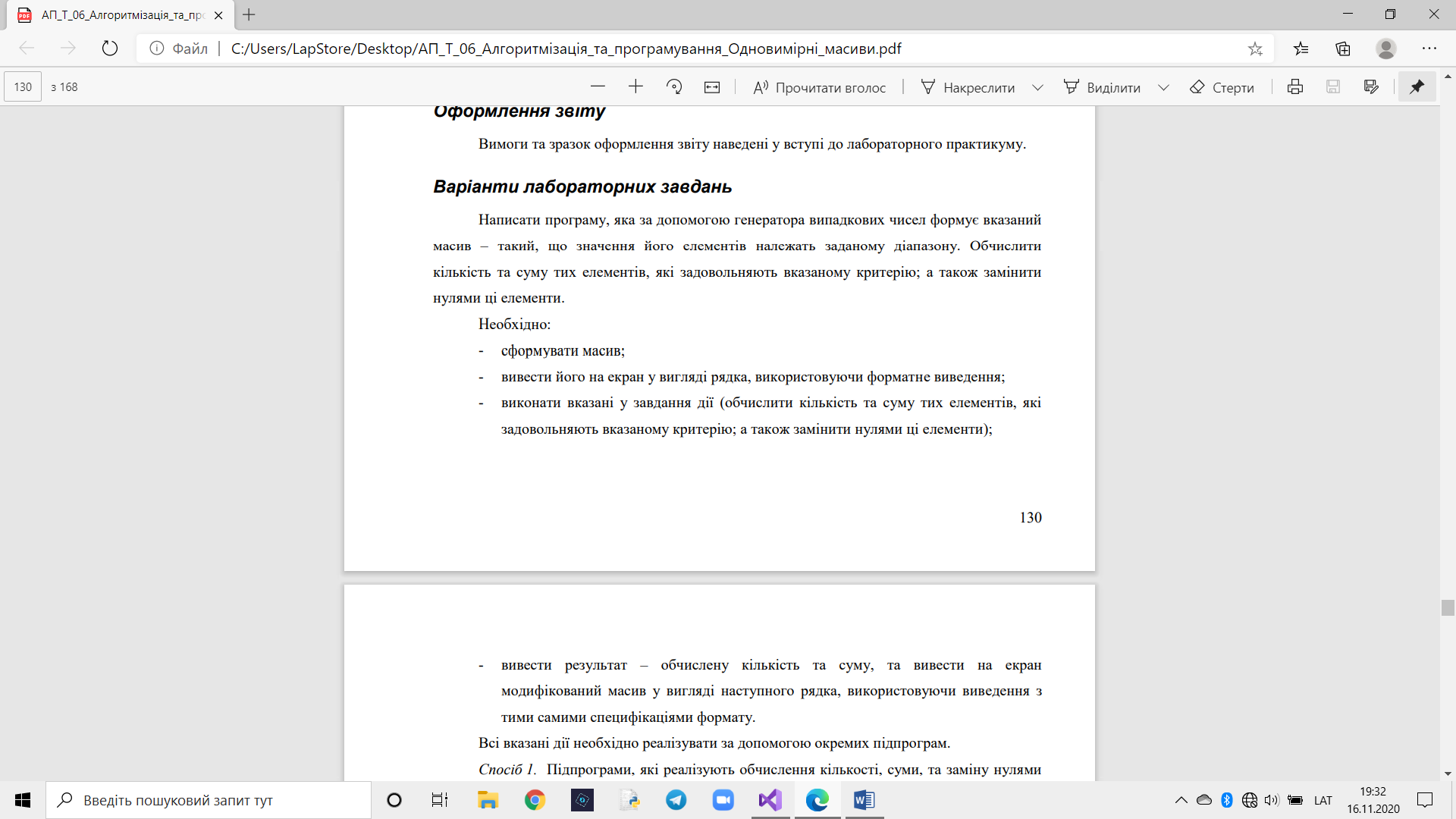
«Пошук елементів одновимірного масиву ітераційним та рекурсивним способом»

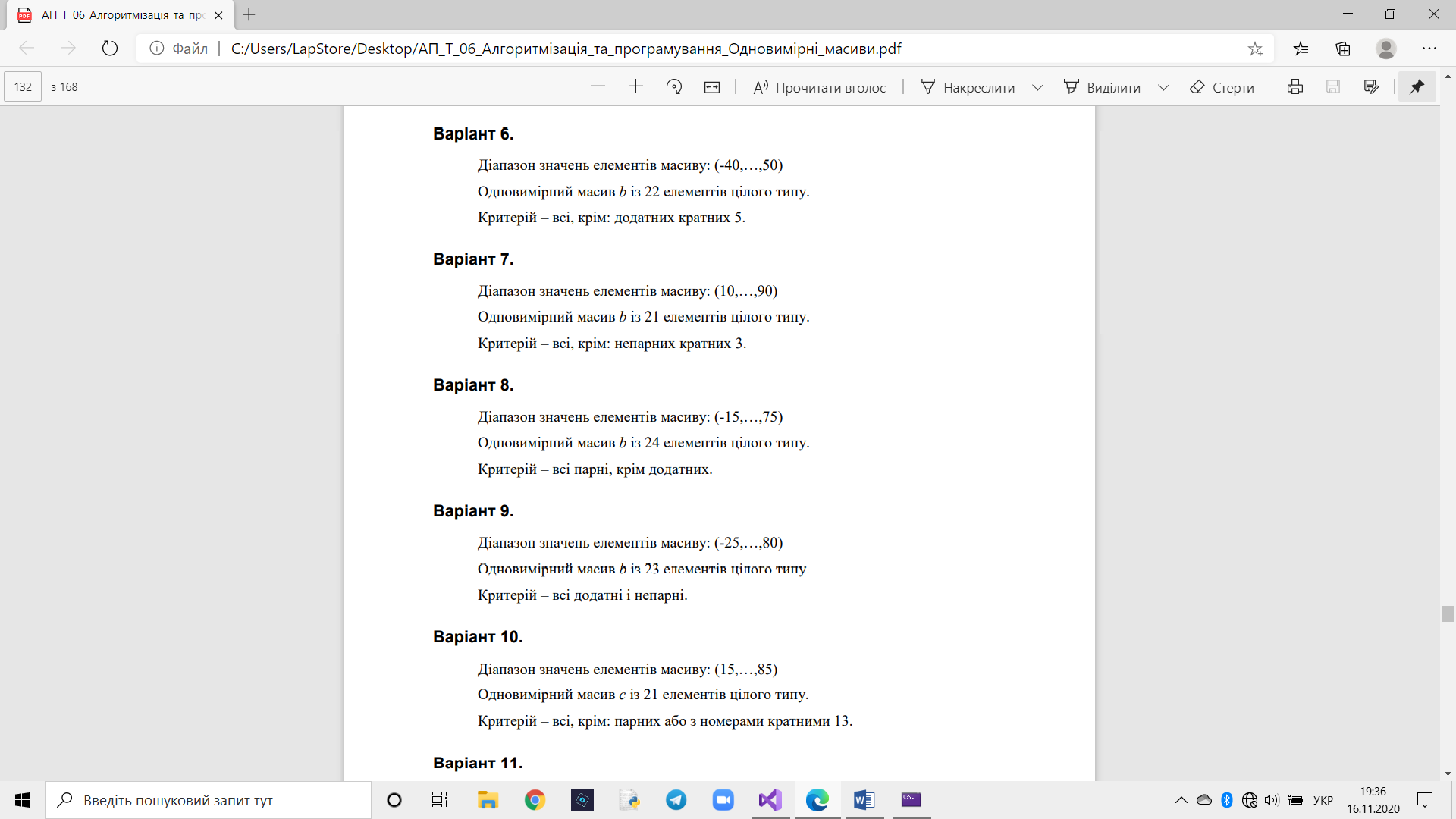
з дисципліни

«Алгоритмізація та програмування»

студента групи ІК-11

Зробка Івана Михайловича





#include <iostream>

using namespace std;

void create(int\* arr, const int n)

{

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

{

arr[i] = rand() % 80 - 25;

}

cout << endl;

}

void print(int\* arr, const int n) {

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

{

cout << arr[i] << " ";

}

cout << endl;

}

int sum(int arr[], int const n) {

int s = 0;

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

{

if (arr[i] > 0 && arr[i] % 2 == 1)

s += arr[i];

}

return s;

}

void count(int arr[], const int n)//згідно варіанту функція виводить всі додатні і непарні елементи масиву.

{

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

{

if (arr[i] > 0 && arr[i] % 2 == 1) {

cout << arr[i] <<" ";

}

}

}

int main()

{

int const n = 23;

int p[n];

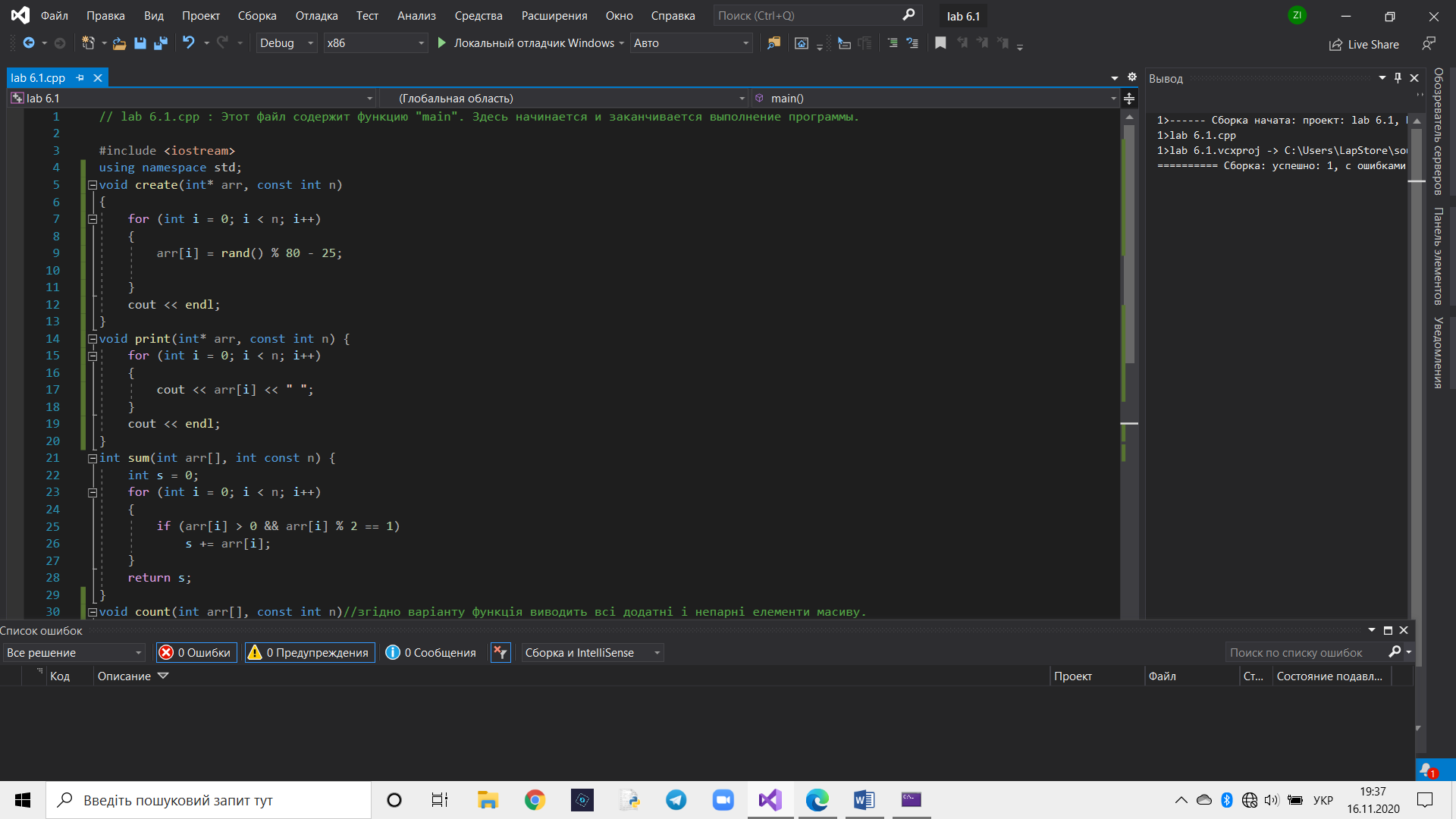
create(p, n);

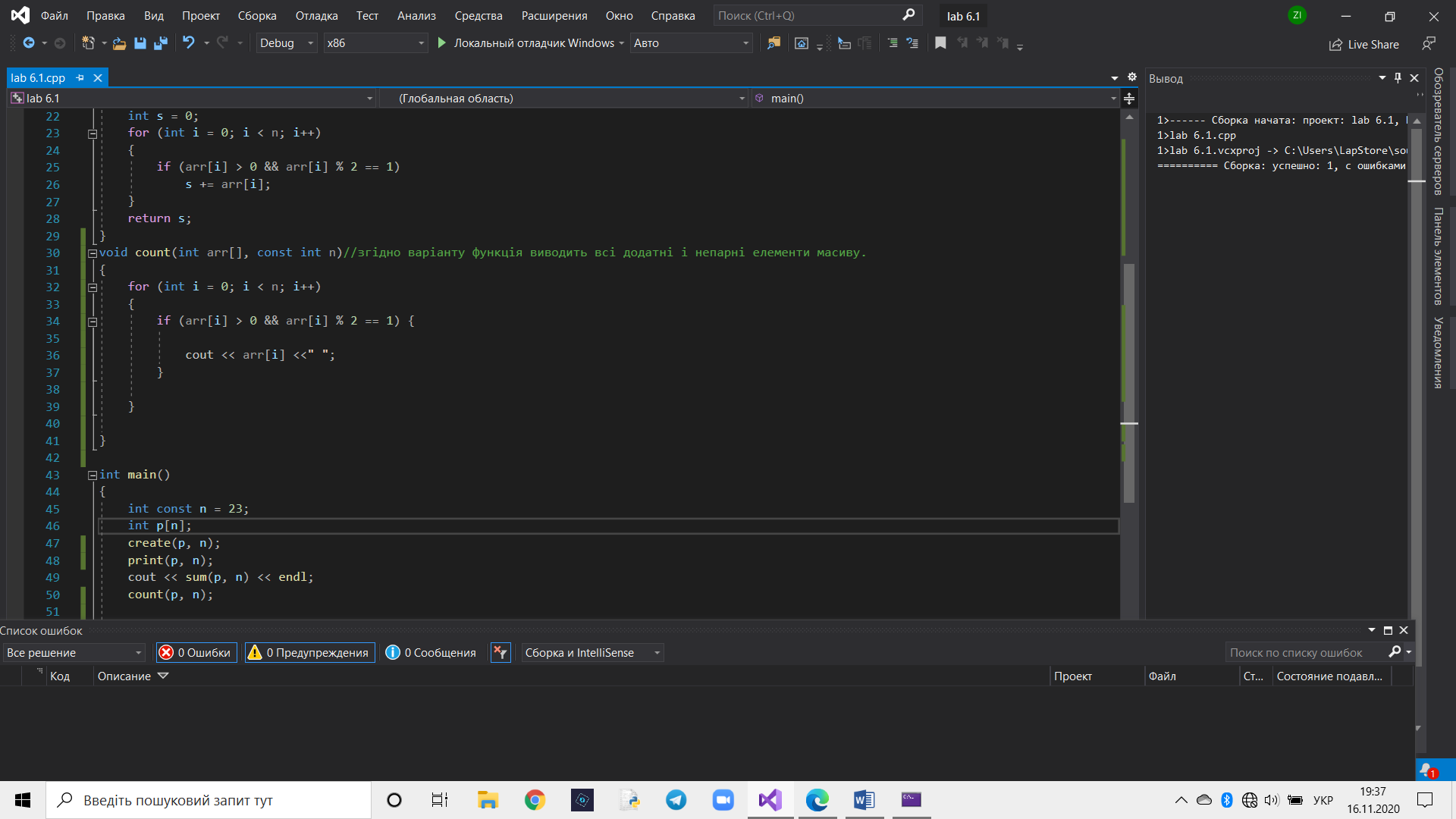
print(p, n);

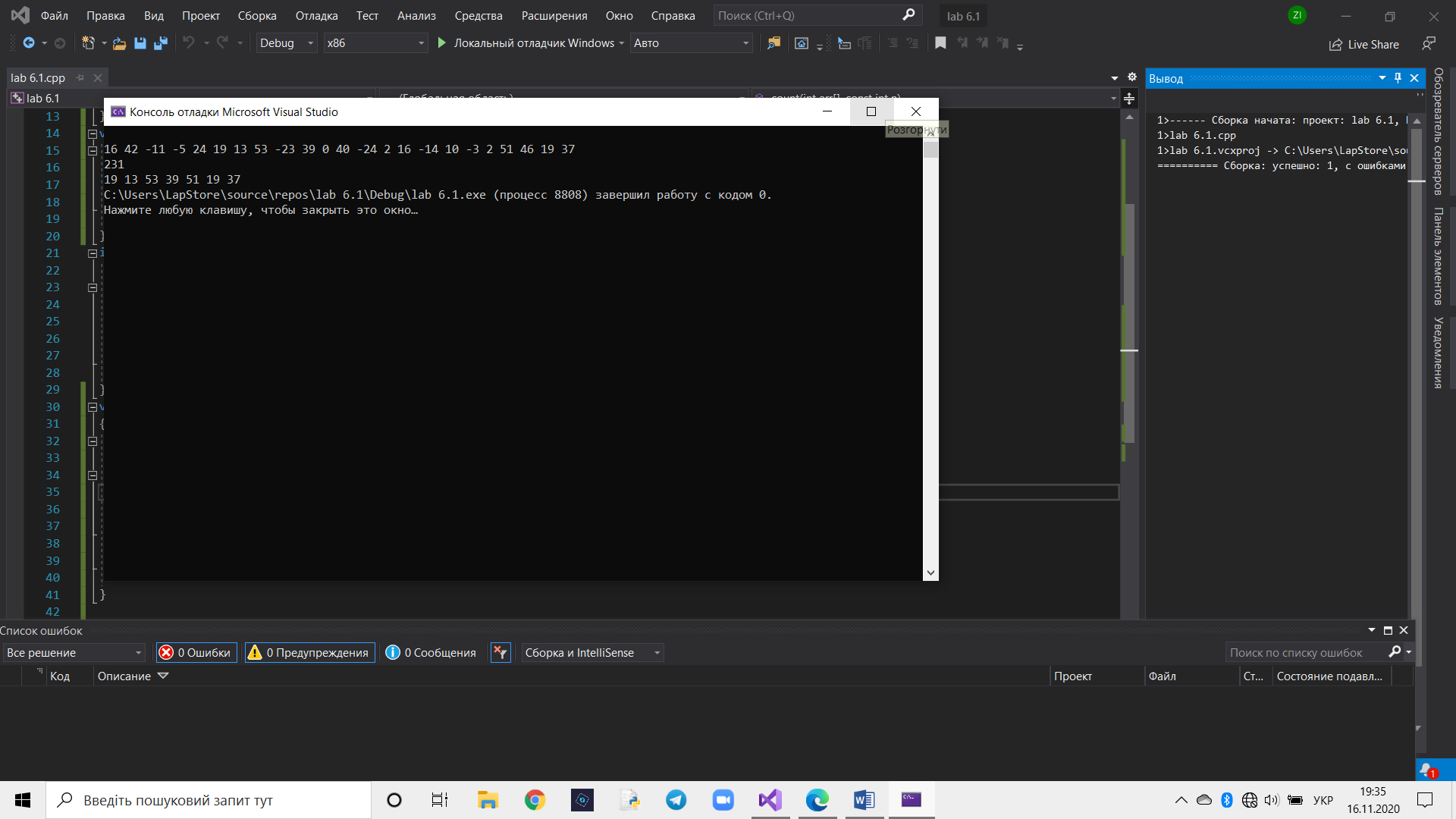
cout << sum(p, n) << endl;

count(p, n);

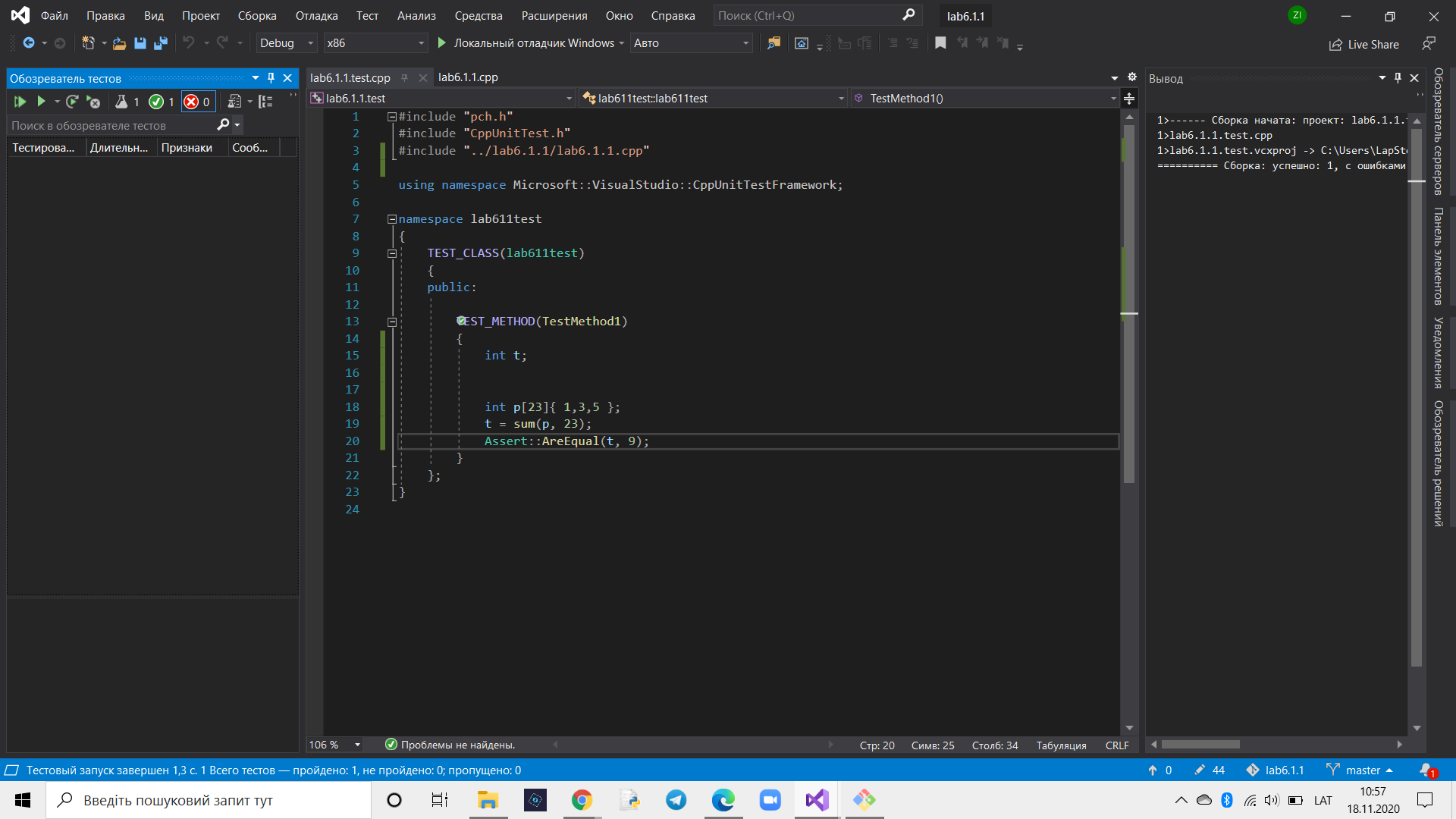
}







**Unit Test**



#include "pch.h"

#include "CppUnitTest.h"

#include "../lab6.1.1/lab6.1.1.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace lab611test

{

TEST\_CLASS(lab611test)

{

public:

TEST\_METHOD(TestMethod1)

{

int t;

int p[23]{ 1,3,5 };

t = sum(p, 23);

Assert::AreEqual(t, 9);

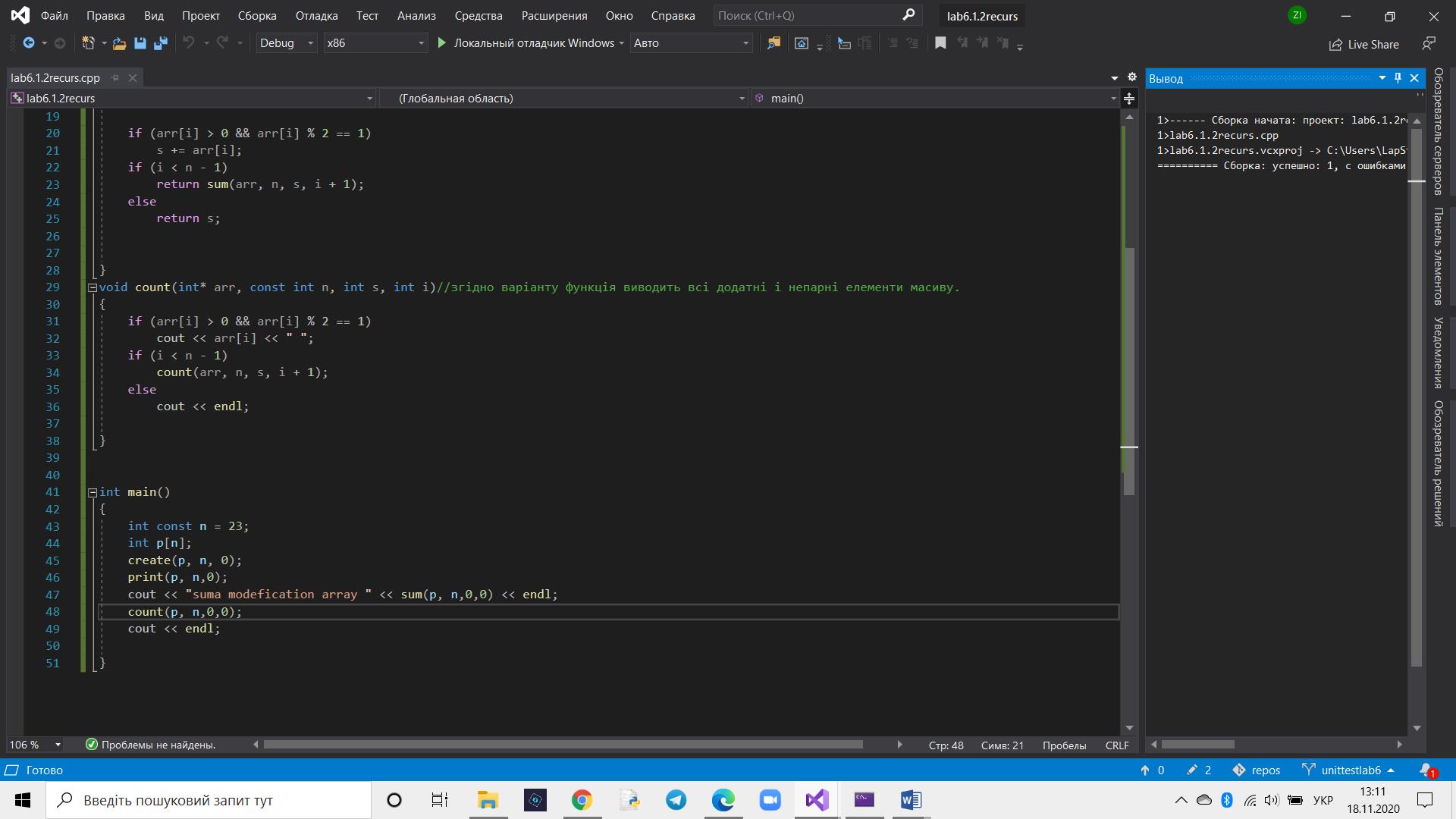
}

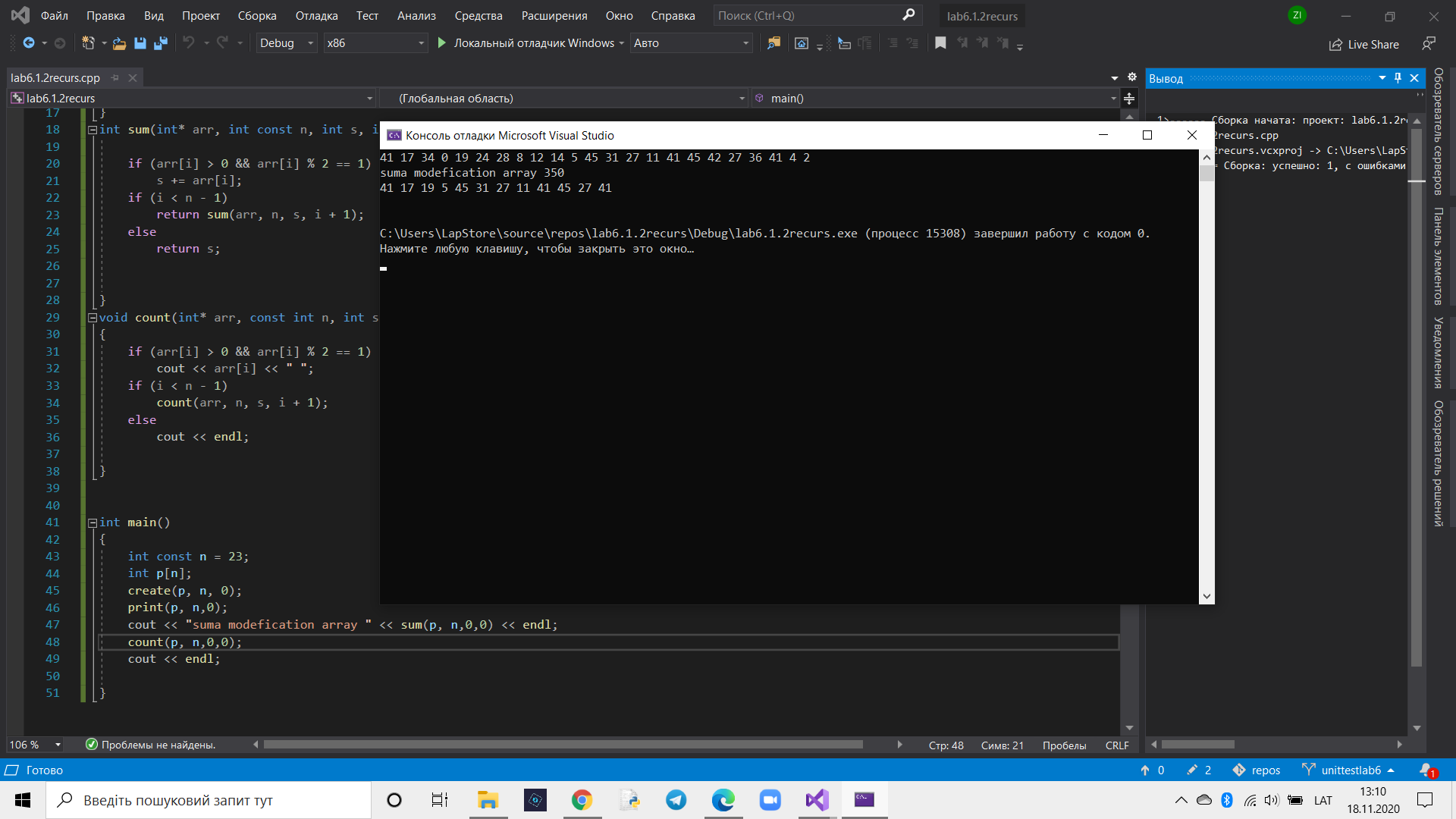
};

}

**Силка на github:** https://github.com/ivanzrobok/lab6.1

**Рекурсивний спосіб**





// lab6.1.2recurs.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

#include <iostream>

using namespace std;

void create(int\* arr, const int n, int i)

{

arr[i] = rand() % 50;

if (i < n - 1)

create(arr, n, i + 1);

}

void print(int\* arr, const int n,int i) {

cout << arr[i]<< " ";

if (i < n - 1)

print(arr, n, i + 1);

else

cout << endl;

}

int sum(int\* arr, int const n, int s, int i) {

if (arr[i] > 0 && arr[i] % 2 == 1)

s += arr[i];

if (i < n - 1)

return sum(arr, n, s, i + 1);

else

return s;

}

void count(int\* arr, const int n, int s, int i)//згідно варіанту функція виводить всі додатні і непарні елементи масиву.

{

if (arr[i] > 0 && arr[i] % 2 == 1)

cout << arr[i] << " ";

if (i < n - 1)

count(arr, n, s, i + 1);

else

cout << endl;

}

int main()

{

int const n = 23;

int p[n];

create(p, n, 0);

print(p, n,0);

cout << "suma modefication array " << sum(p, n,0,0) << endl;

count(p, n,0,0);

cout << endl;

}

Unit Test

#include "pch.h"

#include "CppUnitTest.h"

#include "../lab6.1.2recurs/lab6.1.2recurs.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace lab612recurstest

{

TEST\_CLASS(lab612recurstest)

{

public:

TEST\_METHOD(TestMethod1)

{

int t;

int p[23]{1,7,3};

t = sum(p, 23,0,0);

Assert::AreEqual(t, 11);

}

};

}

