Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЁТ

к лабораторной работе №1

на тему

СТРУКТУРА ПРОГРАММЫ НА СИ.

ФУНКЦИИ ВВОДА-ВЫВОДА

Вариант 15

Проверил Выполнил

асс. каф. ЭВМ студент гр. 318302 П.С. Жуковец В.С. Бахур

МИНСК 2023

**1 ЗАДАНИЕ №1**

* 1. **Условие**

“Ввести длины ребер куба. Найти объем куба и площадь его боковой поверхности. Результат вывести с точностью до трех знаков после запятой.”

* 1. **Алгоритм решения**

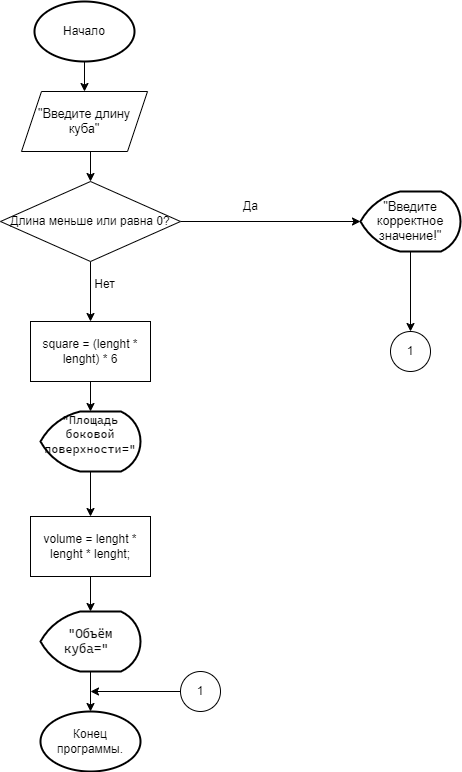


Рисунок 1.1 – Блок-схема

* 1. **Листинг программы**

//Ввести длины ребер куба. Найти объем куба и площадь его боковой поверхности. Результат вывести с точностью до трех знаков после запятой.

#include <stdio.h> //Подкл. библиотеки

#include <locale.h> //Подкл. библиотеки

main() { //Тело программы

setlocale(LC\_ALL, "Rus");

float lenght;

printf("Введите длину куба:"); //вывод текста на экран

scanf\_s("%f", &lenght);

if (lenght <= 0) {

printf("Введите корректное значение!\n");

}

else {

float square;

square = (lenght \* lenght) \* 6;

printf("Площадь боковой поверхности = %.3f\*%.3f\*6 = %.3f\n", lenght, lenght, square); //вывод площади бок. повехности куба

float volume;

volume = lenght \* lenght \* lenght;

printf("Объём куба = %.3f\*%.3f\*.%.3f=%.3f\n", lenght, lenght, lenght, volume);

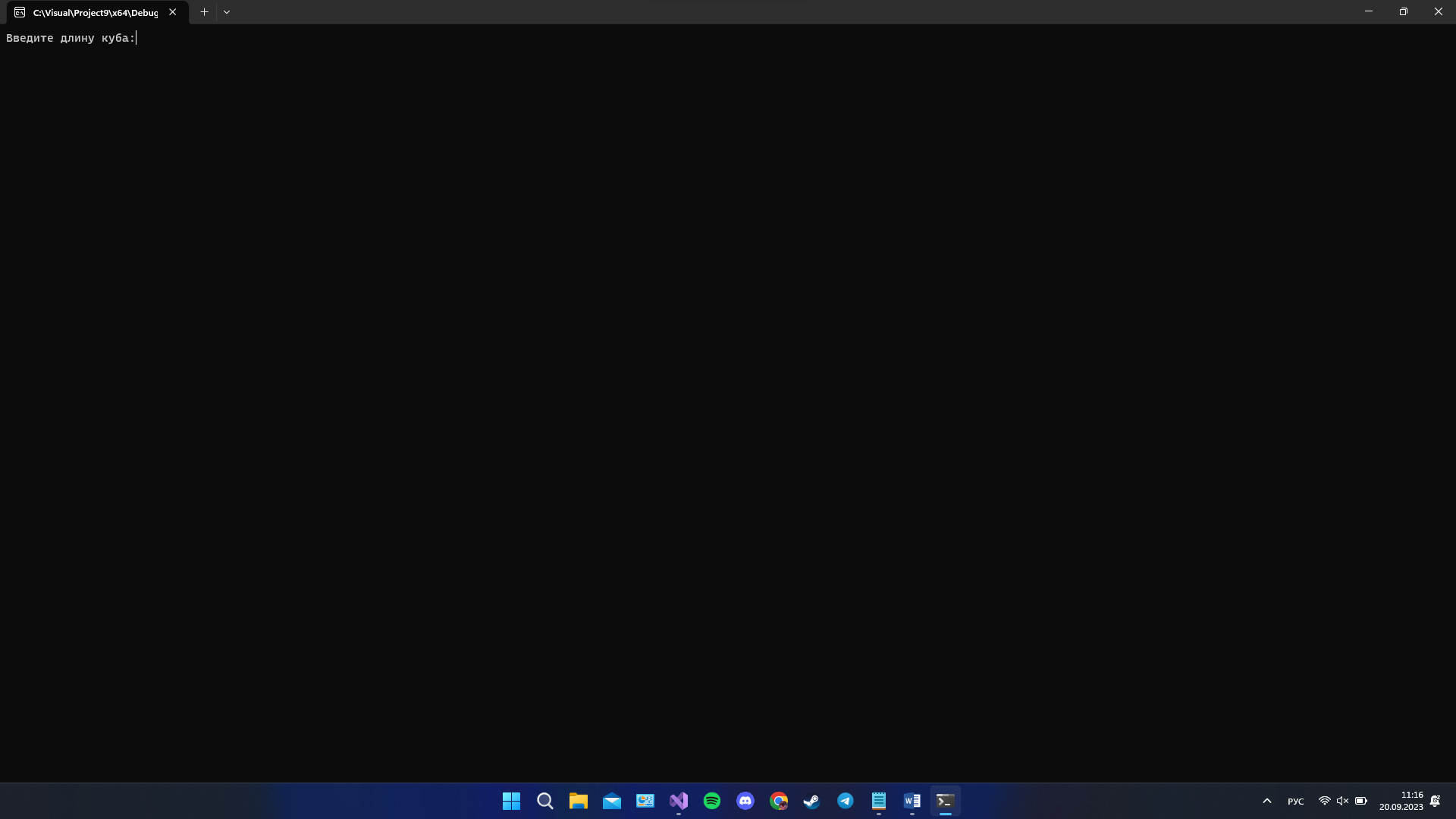
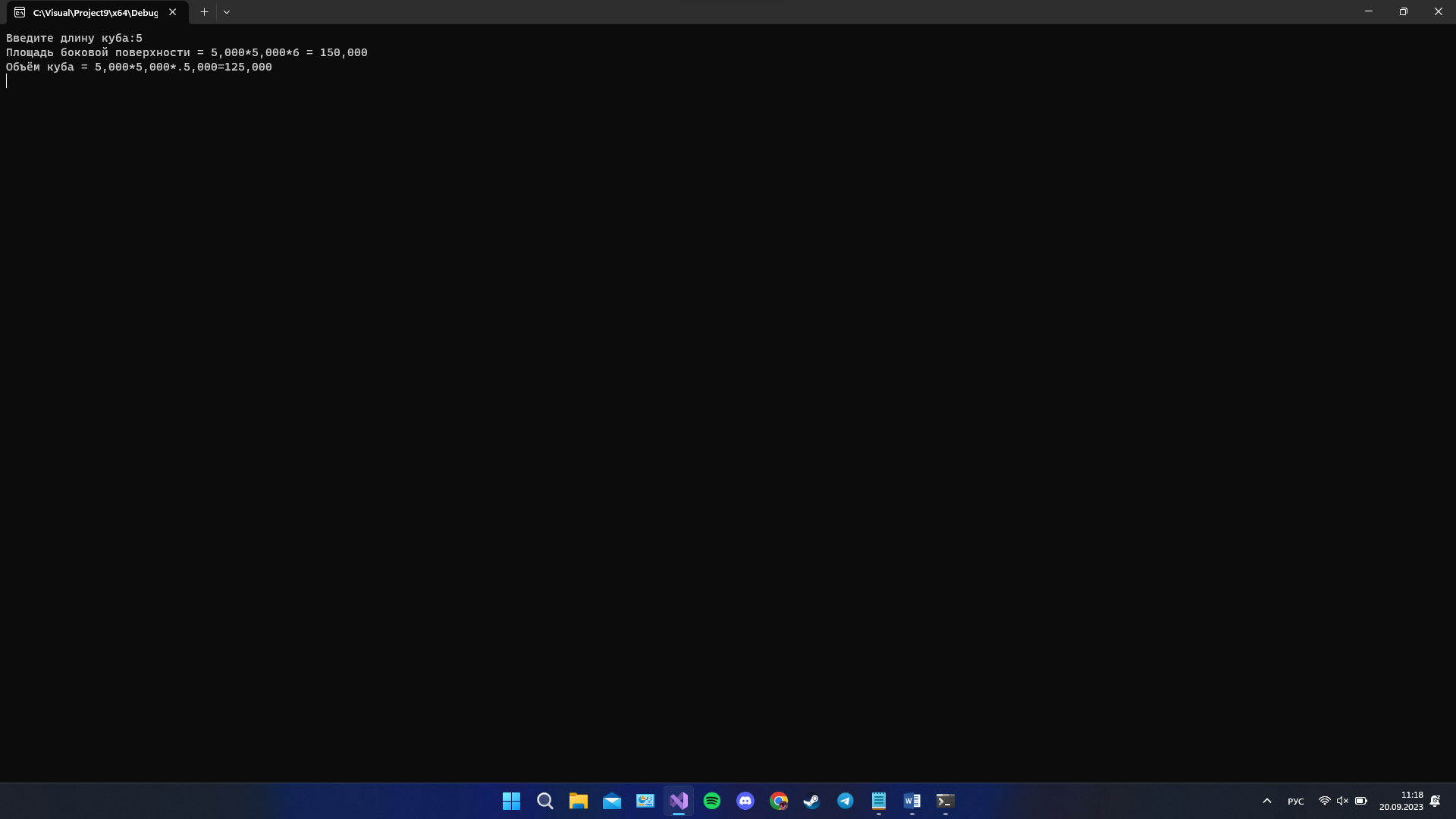
scanf\_s("%f", &volume);

}

}

* 1. **Результат выполнения программы**

Программа предназначена для нахождения площади боковой поверхности и объема куба. Пользователь вводит значение длины стороны куба, программа автоматически делает подсчет и выдает результаты вычисления боковой поверхности и объема куба.

Запрос значений Вывод результата

**2 ЗАДАНИЕ №2**

**2.1 Условие**

“Определить, имеется ли среди целых чисел a, b и c хотя бы одно чётное.”

**2.2 Алгоритм решения**

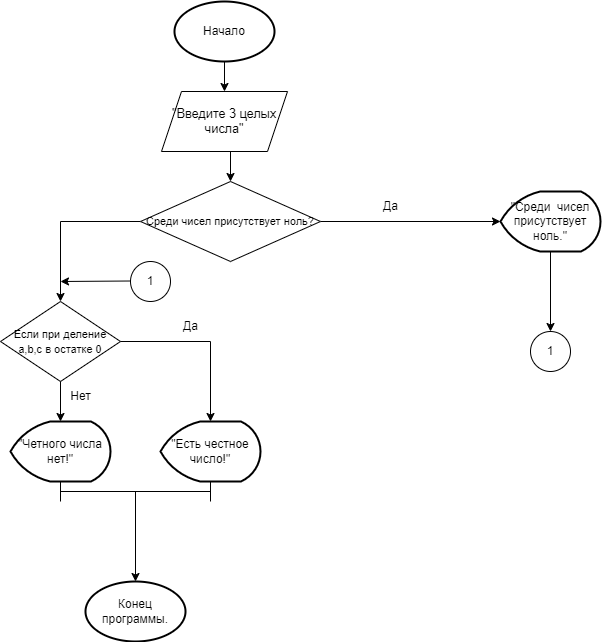
****

Рисунок 2.1 – Блок-схема

**2.3 Листинг программы**

// Определить, имеется ли среди целых чисел a, b и c хотя бы одно чётное.

#include <stdio.h> //Подкл. библиотеки

#include <locale.h> //Подкл. библиотеки для русского языка

main() {

setlocale(LC\_ALL, "Rus"); //Русский язык

int a, b, c;

printf("Введите 3 целых числа:\n");

scanf\_s("%d%d%d", &a, &b, &c);

if ((a == 0) || (b == 0) || (c == 0)) {

printf("Среди чисел присутствует ноль.\n");

}

if ((a % 2 == 0) && (a != 0) || (b % 2 == 0) && (b != 0) || (c % 2 == 0) && (c != 0))

{

printf("Есть четное число!\n");

}

else {

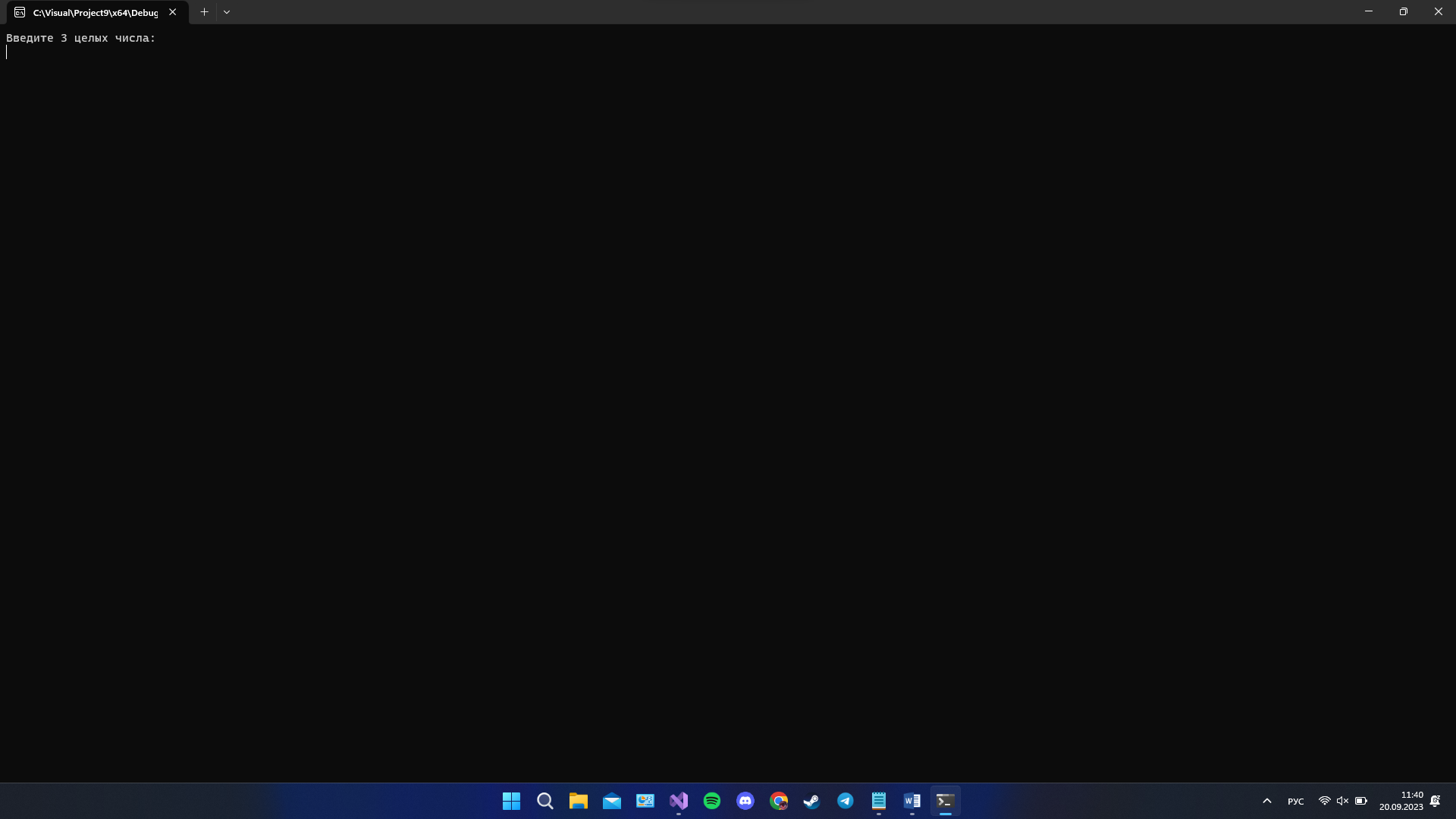
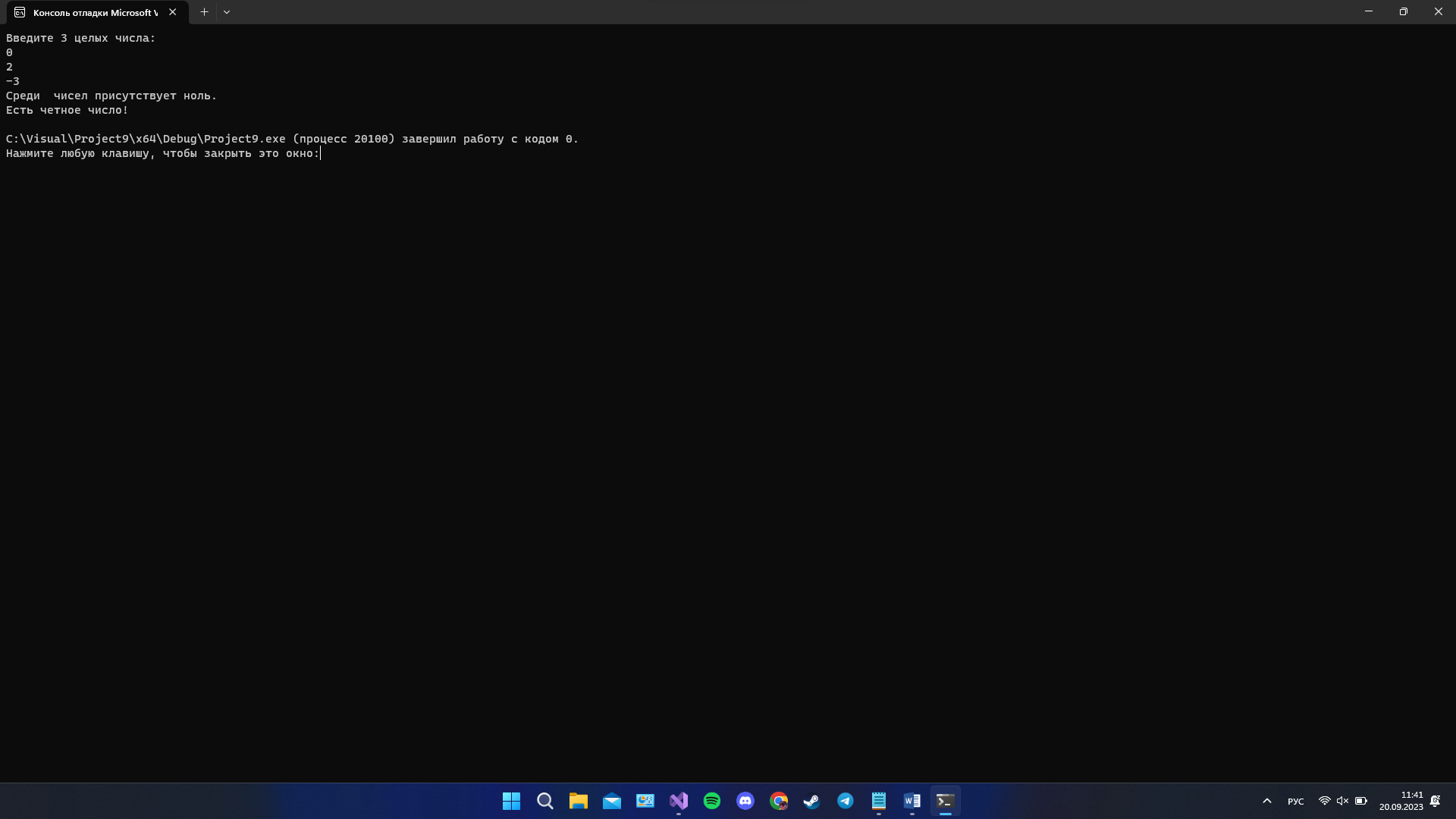
printf("Четного числа нет!\n");

}

}

**2.4 Результат выполнения программы**

Пользователь вводит значение 3-х целых чисел, программа автоматически определяет есть ли четное число среди данных. При наличии четного числа программа вывод на экран надпись: “Есть четное число!”, если четного числа нету: “Четного числа нет!”. (При наличии нуля, программа выводит: “Среди чисел присутствует ноль.”

Запрос чисел Вывод результата

**3 ЗАДАНИЕ №3**

**3.1 Условие**

“ Составьте алгоритм решения неравенства ax2+bx+c<0.”

**3.2 Алгоритм решения**

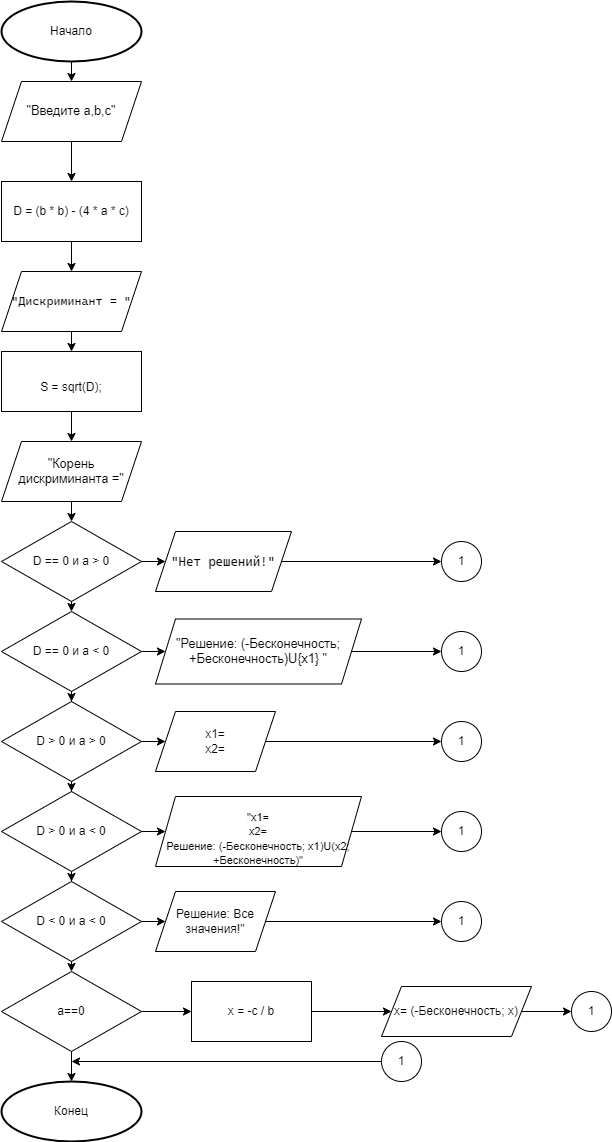


Рисунок 3.1 – Блок-схема

**3.3 Листинг программы**

// Составьте алгоритм решения неравенства ax2+bx+c<0.

#include <stdio.h>

#include <locale.h> //Подкл. библиотеки для русского языка

#include <math.h> //Подкл. бибилиотеку, которая отвечает за математику.

main() {

setlocale(LC\_ALL, "Rus"); //Русский язык

float a, b, c, D, x1, x2, S;

printf("Введите a : ");

scanf\_s("%f", &a);

printf("Введите b : ");

scanf\_s("%f", &b);

printf("Введите c : ");

scanf\_s("%f", &c);

D = (b \* b) - (4 \* a \* c);

printf("Дискриминант = (%f\*%f)-(4\*%f\*%f) = %f\n", b, b, a, c, D);

S = sqrt(D);

printf("Корень дискриминанта = %f\n", S);

if (D == 0 && a > 0)

{

printf("Нет решений!");

}

if (D == 0 && a < 0)

{

x1 = (-b + S) / (2 \* a);

printf("Решение: (-Бесконечность; +Бесконечность)U{x1} = %f", x1);

}

if (D > 0 && a > 0)

{

x1 = (-b + S) / (2 \* a);

x2 = (-b - S) / (2 \* a);

printf("x1= %f\n", x1);

printf("x2= %f ", x2);

}

if (D > 0 && a < 0)

{

x1 = (-b + S) / (2 \* a);

x2 = (-b - S) / (2 \* a);

printf("x1= %f\n", x1);

printf("x2= %f\n ", x2);

printf("Решение: (-Бесконечность; %f)U(%f; +Бесконечность)", x1, x2);

}

if (D < 0 && a < 0)

{

printf(" Решение: Все значения!");

}

if (a == 0)

{

double x = 0;

x = -c / b;

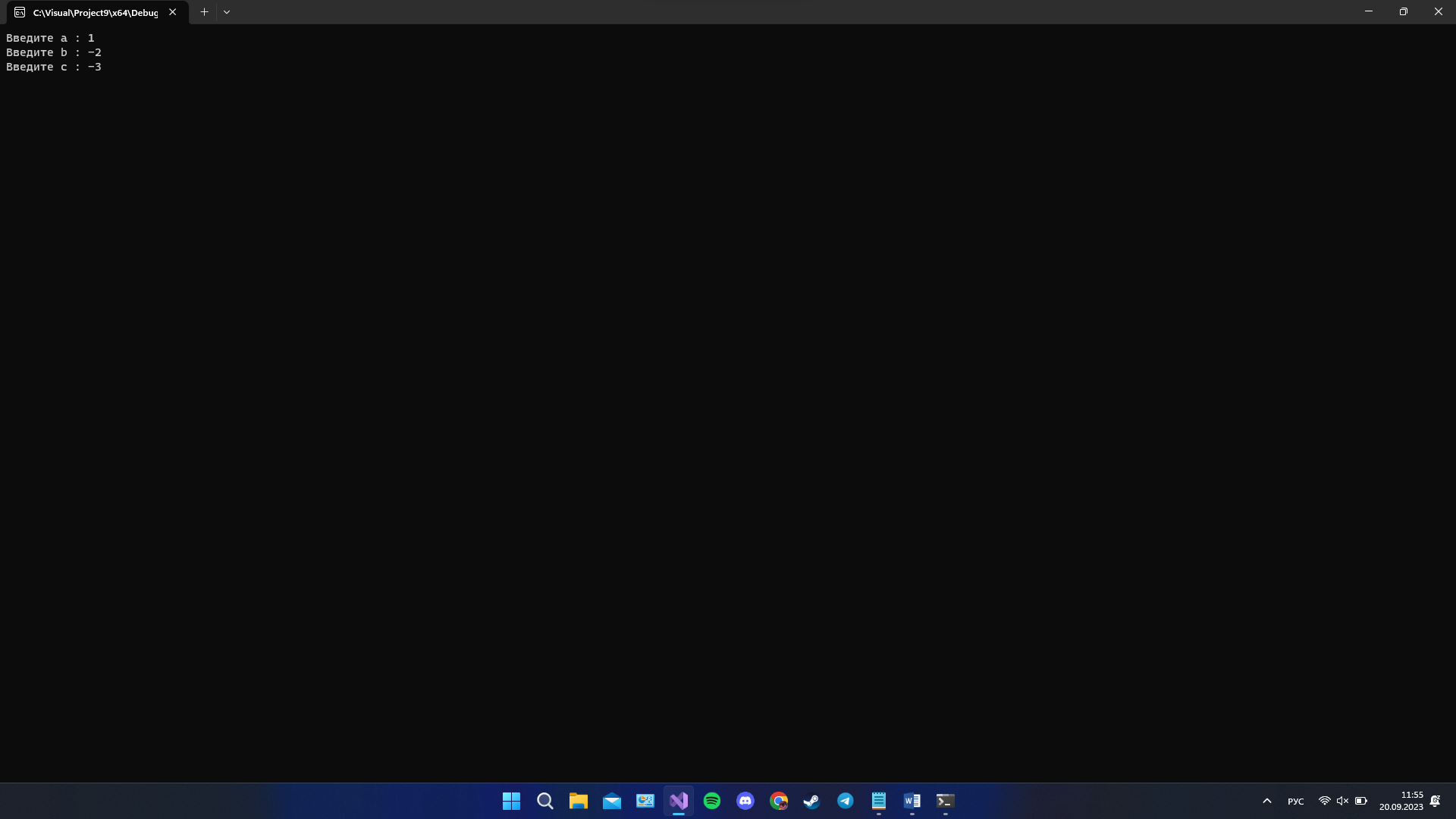
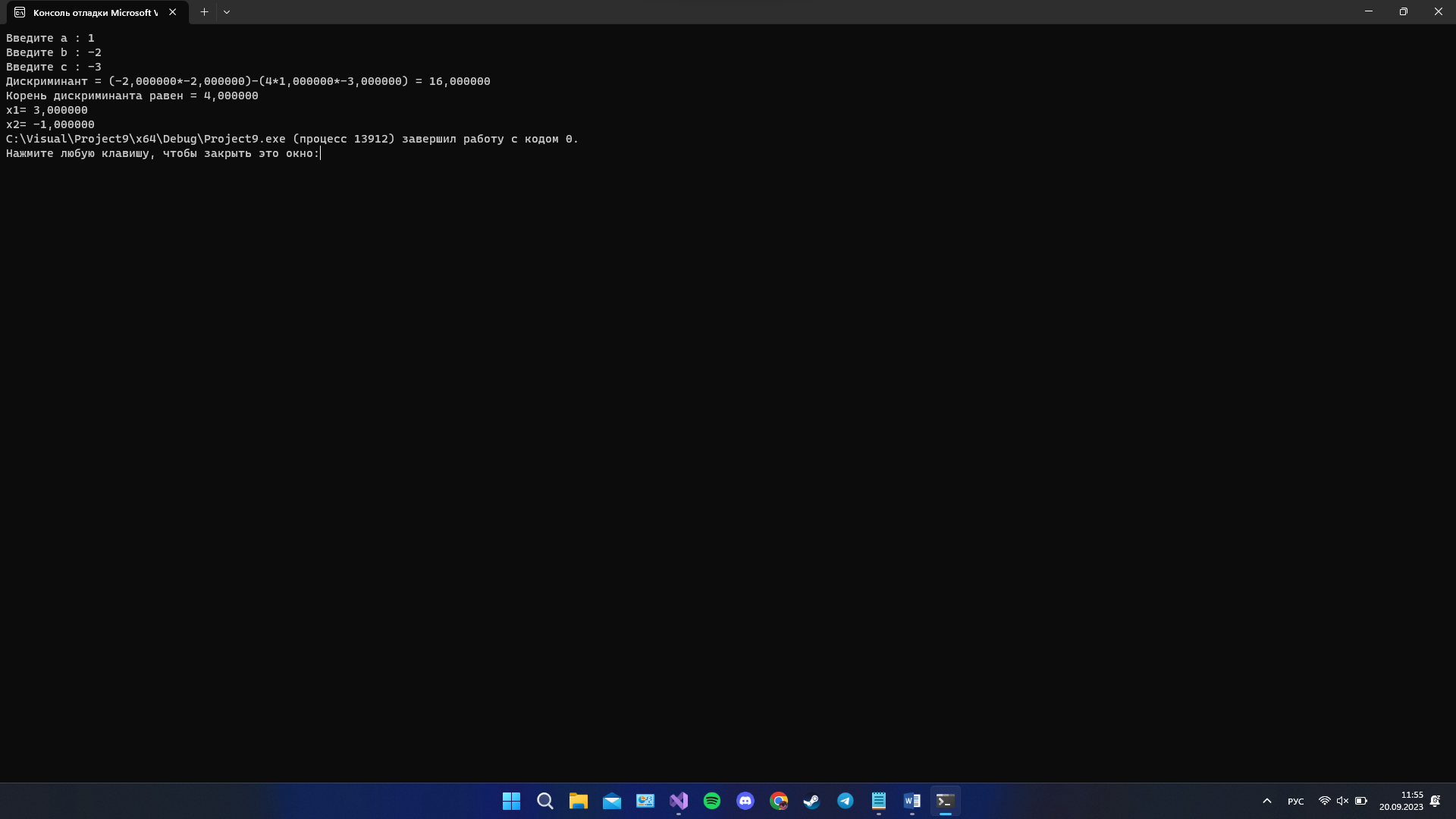
printf("x= %f\n(-Бесконечность; %f)", x,x);

}

}

**3.4 Результат выполнения программы**

Пользователь вводит значение 3-х переменных (a, b, c), программа автоматически считает дискриминант и находить все решения неравенства (выводя на экран «корни» (х1, х2) и промежутки, при которых неравенство верное) или доказывает, что их нет.

Ввод данных Вывод результата

**4 ЗАДАНИЕ №4**

**4.1 Условие**

“Для натурального числа К напечатать фразу "мы нашли К грибов в лесу", согласовав окончание слова "гриб" с числом К.”

**4.2 Алгоритм решения**

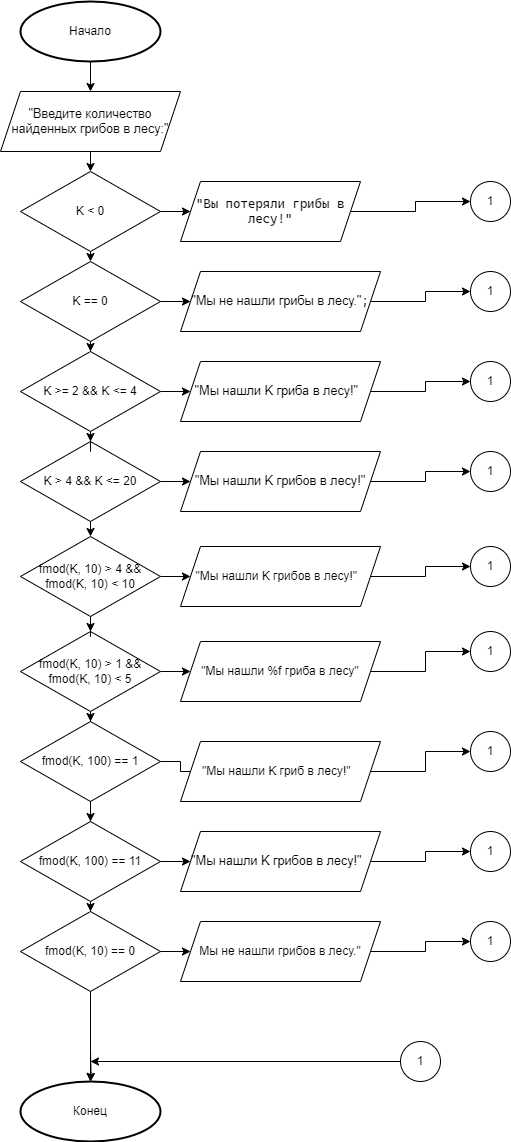
****

Рисунок 3.1 – Блок-схема

**4.3 Листинг программы**

// Для натурального числа К напечатать фразу "мы нашли К грибов в лесу", согласовав окончание слова "гриб" с числом К.

//fmod отсаток от деления

#include <stdio.h>

#include<locale.h>

#include <math.h>

main() {

setlocale(LC\_ALL, "Rus"); //Русский язык

float K;

printf("Введите количество найденных грибов в лесу:");

scanf\_s("%f", &K);

if (K < 0) {

printf("Вы потеряли грибы в лесу!");

}

else if (K == 0) {

printf("Мы не нашли грибы в лесу.");

}

else {

if (K == 1) {

printf("Мы нашли %f гриб в лесу!", K);

}

else if (K >= 2 && K <= 4) {

printf("Мы нашли %f гриба в лесу!", K);

}

else if (K > 4 && K <= 20) {

printf("Мы нашли %f грибов в лесу!", K);

}

else if (fmod(K, 10) > 4 && fmod(K, 10) < 10) {

printf("Мы нашли %f грибов в лесу!", K);

}

else if (fmod(K, 10) > 1 && fmod(K, 10) < 5) {

printf("Мы нашли %f гриба в лесу!", K);

}

else if (fmod(K, 100) == 1) {

printf("Мы нашли %f гриб в лесу!", K);

}

else if (fmod(K, 100) == 11) {

printf("Мы нашли %f грибов в лесу!", K);

}

else if (fmod(K, 10) == 0) {

printf("Мы не нашли грибов в лесу.");

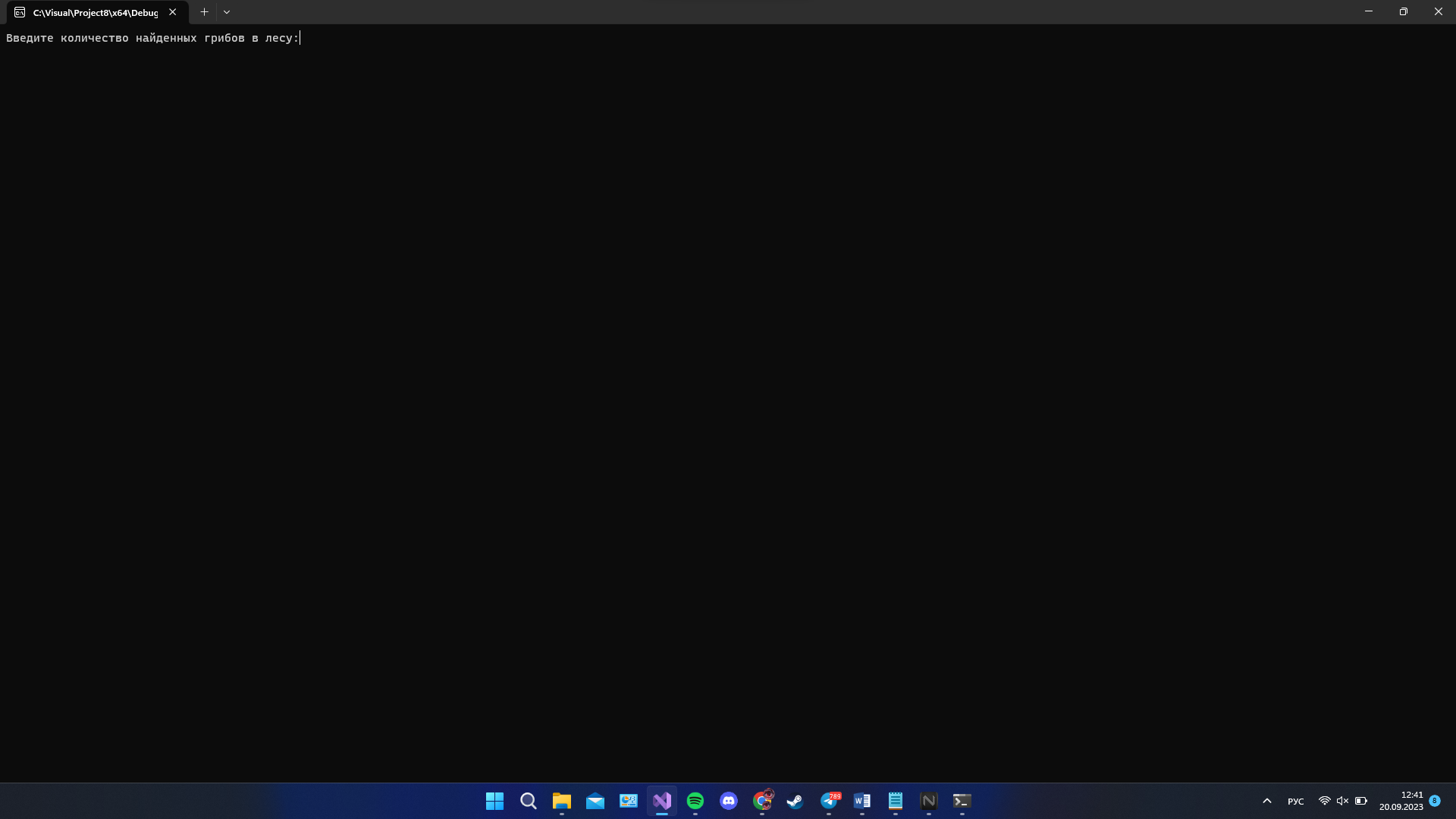
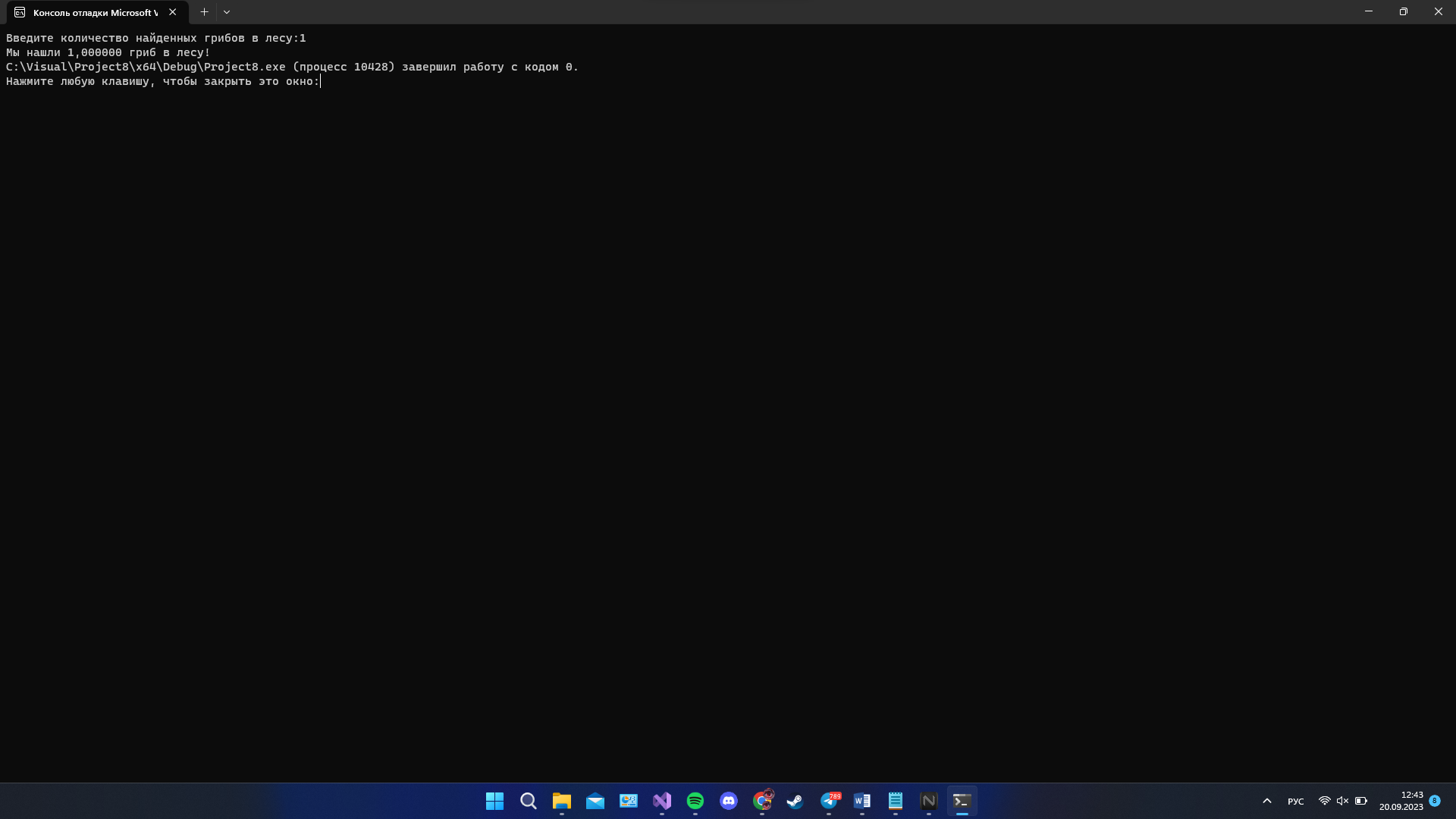
}

}

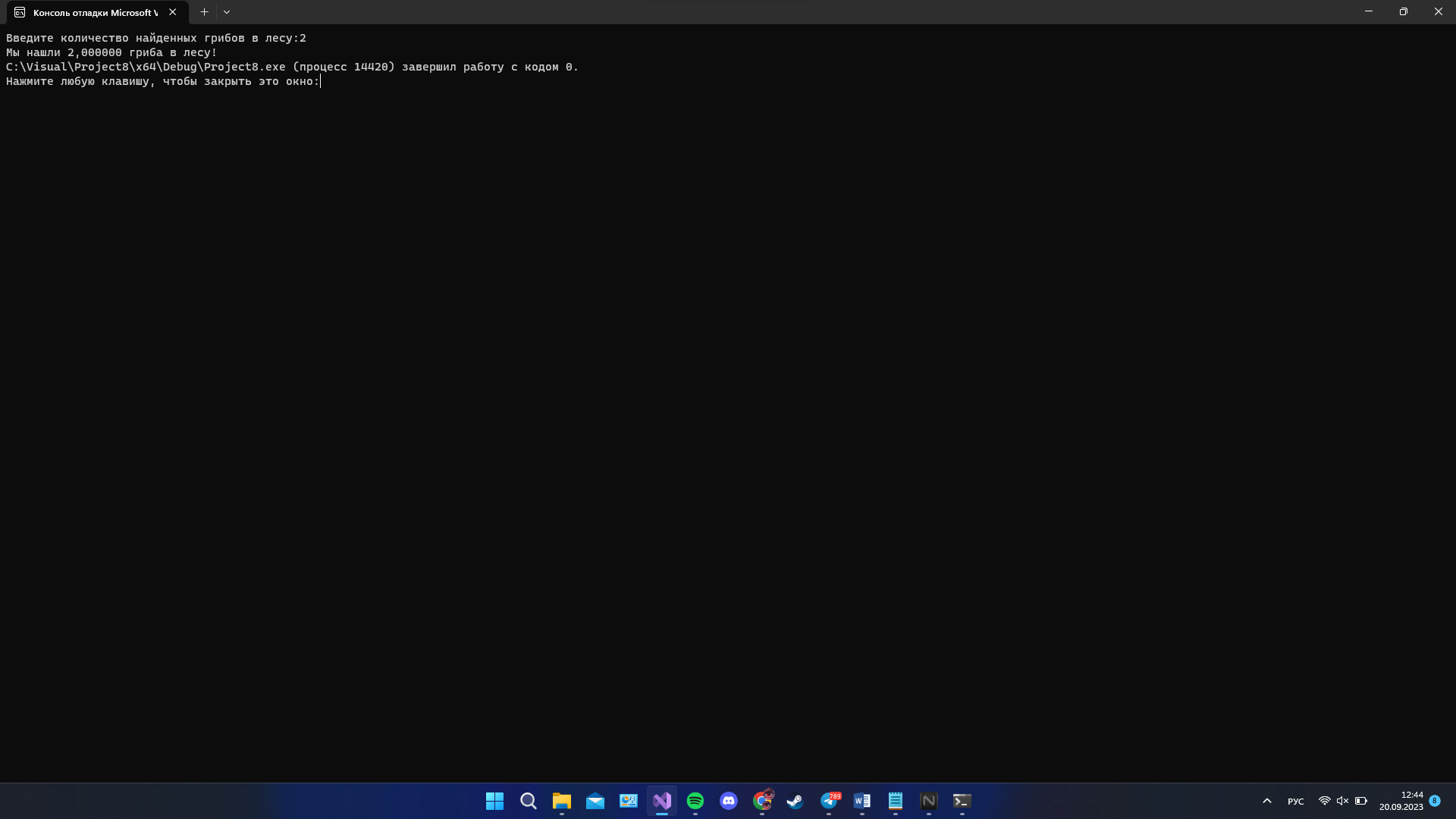
}

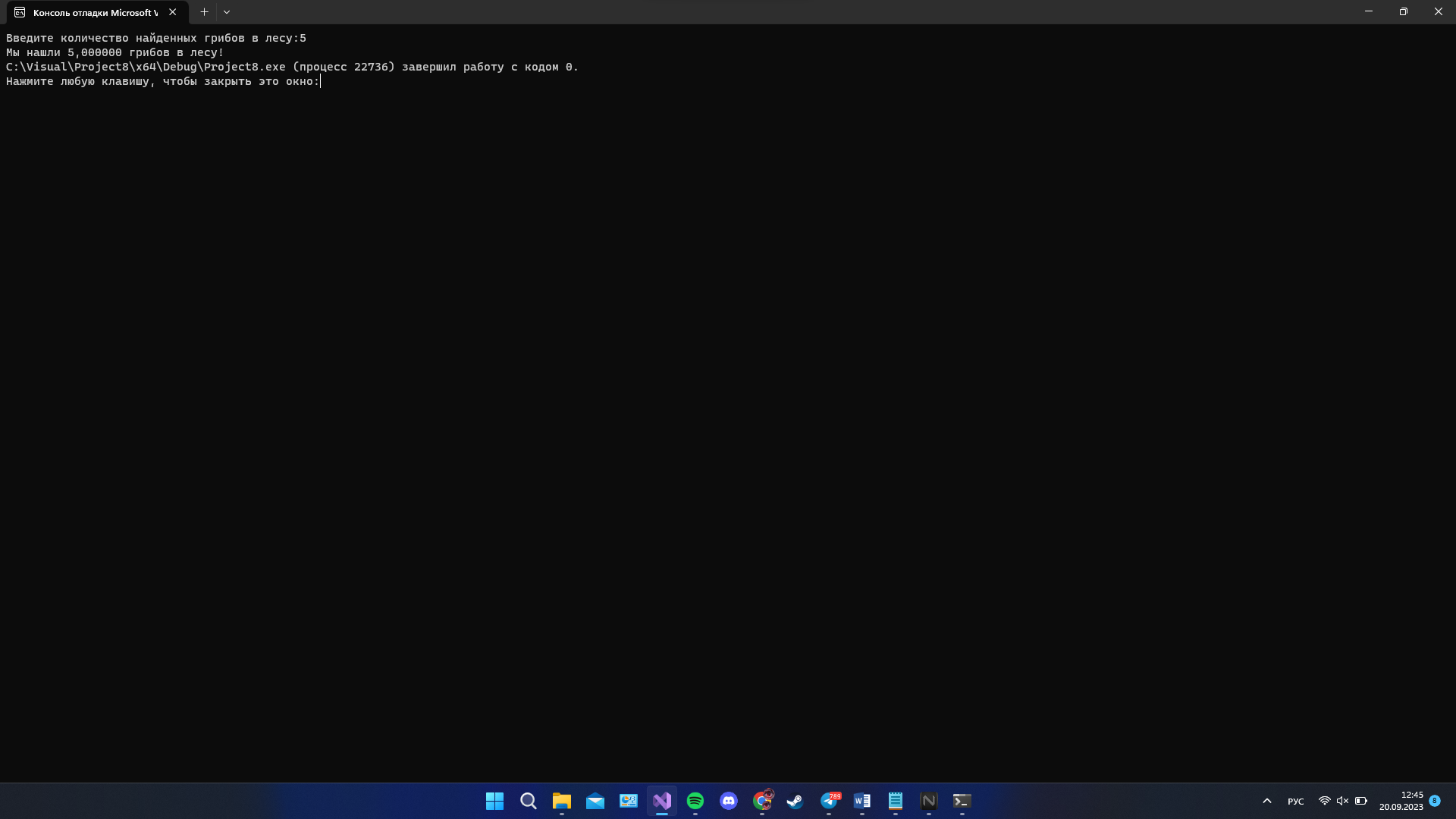
**4.4 Результат выполнения программы**

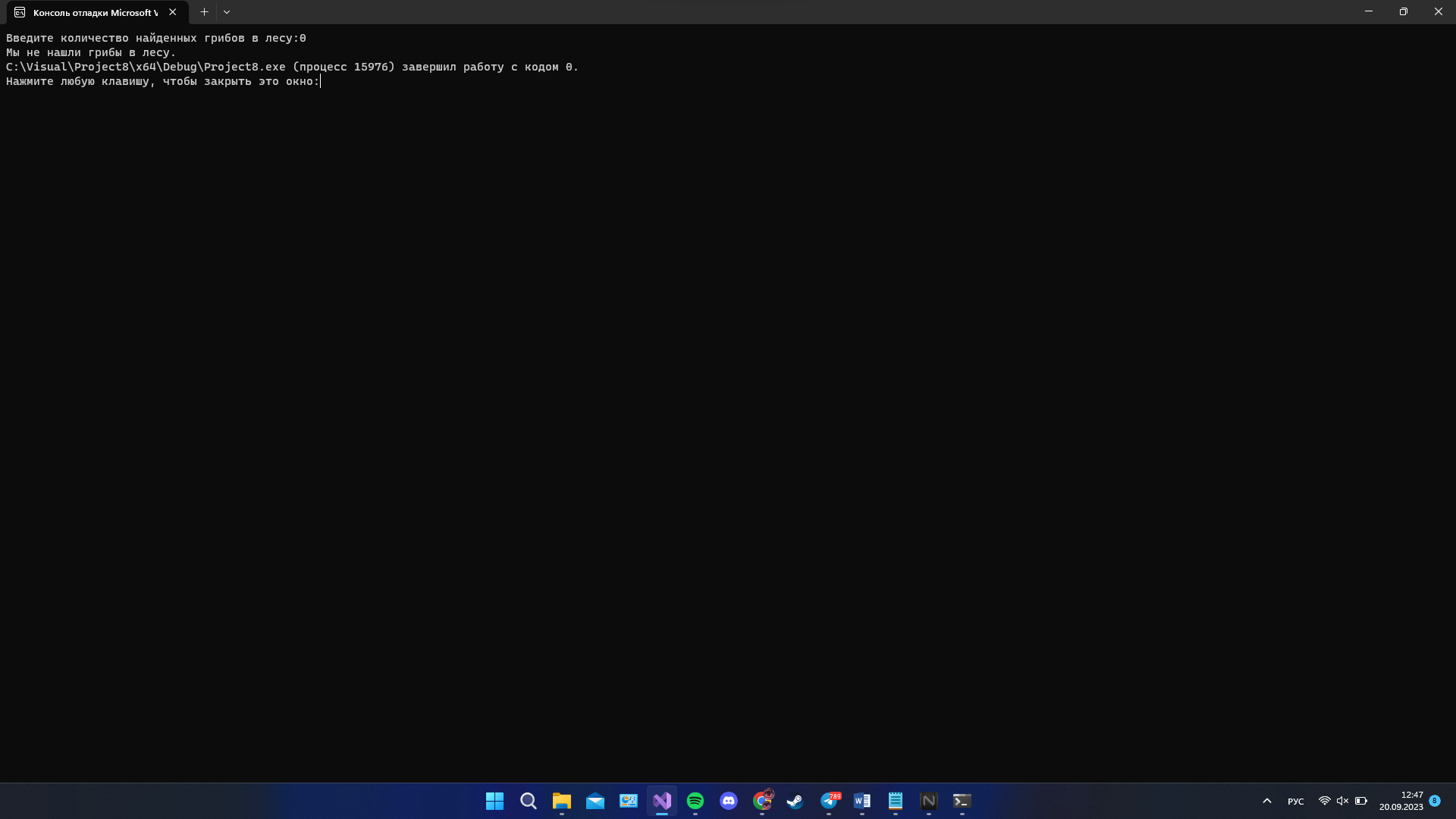
Пользователь вводит значение найденных грибов, программа автоматически согласует окончание слова «гриб» с их числом и выведет результат на экран в виде: “Мы нашли 1 гриб в лесу!”; “Мы нашли 2 гриба в лесу!”; “Мы нашли 5 грибов в лесу!” и т.д. При попытке ввода 0 в виде найденных грибов, программа выводит на экран: “Мы не нашли грибы в лесу.”, при вводе отрицательного количества грибов программа выводит: “Вы потеряли грибы в лесу.”

Запрос значения (-ий)







Вывод результата