**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДИРАЦИИ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования**

**Московский технический университет связи и информатики**

**(МТУСИ)**

Направление 11.03.02 Инфокоммуникационные технологии

и системы связи

Профиль подготовки Программно-защищенные инфокоммуникации

Факультет ИБ

Кафедра СИТиС

Вариант 27

Предмет Объектно-ориентированное программирование систем защиты

**Регулярные и итерационные циклы**

Выполнила: студентка группы БП31801 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Яковлева К.А. Проверил: ст.преподаватель каф.ИБ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Барков В.В.

Москва 2021

**Цель работы**

Овладеть навыками создания циклических алгоритмов на языке C++ с применением инструкций цикла на примере алгоритмов работы с последовательностями чисел.

Все указанные ниже функции разработать в трёх вариантах: с использованием цикла for, while, do … while. Функциям из задания дать одинаковые имена. Каждую функцию поместить в отдельный файл. Прототип функции вынести в заголовочный файл. В решении создать три проекта.

**Задание**

Индивидуальное:

Общее:

**Задание 1.**

Написать функцию для вычисления выражения 

Функция в качестве параметра принимает значение 𝑛 и вычисляет сумму 𝑛 слагаемых.

Прототип функции double summ(int n).

Функция не должна использовать функции консольного ввода-вывода.

**Задание 2.**

Написать функцию для вычисления выражения  с точностью 𝜀.

Функция в качестве параметра принимает значение 𝜀 и вычисляет сумму до тех пор, пока не выполнится условие |𝑎𝑖|≤𝜀.

Прототип функции double summ2(double eps).

Функция не должна использовать функции консольного ввода-вывода.

**Задание 3.**

Написать функцию, которая печатает 𝑛 членов последовательности {𝑎𝑖}, исключая из неё каждый n-ый член.

Числа 𝑛 и 𝑘 передаются в функцию в качестве параметра. Для исключения члена последовательности используйте инструкцию перехода continue.

Прототип функции void print(int n, int k).

Пример вывода: 1.2 1.3 0.75

**Задание 4.**

Написать функцию, которая возвращает номер первого члена последовательности {𝑎𝑖}, для которого выполняется условие |𝑎𝑖|≤𝜀.

Функция в качестве параметра принимает значение 𝜀.

Выход из цикла организовать с помощью инструкции перехода break.

Прототип функции int findFirstElement(double eps).

Функция не должна использовать функции консольного ввода-вывода.

**Задание 5.**

Написать функцию, которая возвращает номер первого отрицательного члена последовательности {𝑎𝑖}, для которого выполняется условие |𝑎𝑖|≤𝜀.

Функция в качестве параметра принимает значение 𝜀.

Выход из цикла организовать с помощью инструкции перехода return.

Прототип функции int findFirstNegativeElement(double eps).

Функция не должна использовать функции консольного ввода-вывода.

**Задание 6.**

Разработать функцию main(), которая в цикле выводит на экран меню из 6 пунктов и ожидает ответа пользователя:

1. Задание 1

2. Задание 2

3. Задание 3

4. Задание 4

5. Задание 5

6. Выход

Обработку пользовательского ввода выполнить с помощью инструкции switch. При нажатии на соответствующую цифру меню стирается с экрана и приложение переходит в режим демонстрации работы выбранного задания. По завершении демонстрации программа ожидает пользовательского ввода для перехода в меню. Выход из программы должно осуществляться при нажатии цифры 6 в режиме меню. Скомпилировать 3 версии программы:

1) Все функции реализованы с помощью цикла for

2) Все функции реализованы с помощью цикла while

3) Все функции реализованы с помощью цикла do … while

**Выполнение с циклом For**

Листинг 1 – прототип функции double summ(int n) с использованием цикла For.

double summ(int n)

{

double a = 0;

double sum = 0;

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

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

cout << "n=" << i << " " << "f=" << a << "\n";

}

return sum;

}

Листинг 2 – прототип функции double summ2(double eps) с использованием цикла For.

double summ2(double eps)

{

int i = 0;

double a = 0;

double sum = 0;

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

for (int i = 0; abs(a) > eps; i++)

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

cout << "n=" << i << " " << "f=" << a << "\n";

}

return sum;

}

Листинг 3 – прототип функции void print(int n, int k) с использованием цикла For.

void print(int n, int k)

{

double count = 0;

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

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if ((i + 1) % k == 0)

{

continue;

}

else

{

cout << count << " ";

}

}

}

Листинг 4 – прототип функции int findFirstElement(double eps) с использованием цикла For.

int findFirstElement(double eps)

{

int i;

double count = 0;

for (i = 0; ; i++)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if (abs(count) < eps)

{

break;

}

}

return i + 1;

}

Листинг 5 – прототип функции int findFirstNegativeElement(double eps) с использованием цикла For.

int findFirstNegativeElement(double eps)

{

int i;

double count = 0;

for (i = 0; ; i++)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if (abs(count) < eps && count < 0)

{

return i + 1;

}

}

}

Листинг 6 – функция main() с использованием цикла For.

#include "pch.h"

#include <iostream>

using namespace std;

void taskk1(), task2(), task3(), task4(), task5();

void menu()

{

cout << "Задания ЛР3 с циклом For" << endl

<< "[1] Задание 1 " << endl

<< "[2] Задание 2" << endl

<< "[3] Задание 3" << endl

<< "[4] Задание 4" << endl

<< "[5] Задание 5" << endl

<< "[6] Выход" << endl

<< "Введите номер подпункта: ";

}

int main()

{

int g;

setlocale(LC\_CTYPE, "rus");

while (true)

{

menu();

cin >> g;

system("cls");

switch (g)

{

case 1:

taskk1();

break;

case 2:

task2();

break;

case 3:

task3();

break;

case 4:

task4();

break;

case 5:

task5();

break;

case 6:

return true;

}

system("pause");

}

}

Листинг 7 – файл Source.cpp

#include "pch.h"

#include <iostream>

using namespace std;

double summ(int n)

{

double a = 0;

double sum = 0;

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

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

cout << "n=" << i << " " << "f=" << a << "\n";

}

return sum;

}

double summ2(double eps)

{

int i = 0;

double a = 0;

double sum = 0;

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

for (int i = 0; abs(a) > eps; i++)

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

cout << "n=" << i << " " << "f=" << a << "\n";

}

return sum;

}

void print(int n, int k)

{

double count = 0;

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

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if ((i + 1) % k == 0)

{

continue;

}

else

{

cout << count << " ";

}

}

}

int findFirstElement(double eps)

{

int i;

double count = 0;

for (i = 0; ; i++)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if (abs(count) < eps)

{

break;

}

}

return i + 1;

}

int findFirstNegativeElement(double eps)

{

int i;

double count = 0;

for (i = 0; ; i++)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if (abs(count) < eps && count < 0)

{

return i + 1;

}

}

}

void taskk1()

{

int i;

cout << "Введите число n\nПрограмма вычислит сумму n слагаемых\n";

cin >> i;

cout << "Сумму sum = " << summ(i);

}

void task2()

{

double eps;

cout << "Программа вычислит с заданной точностью \nВведите eps = ";

cin >> eps;

cout << "Function summ = " << summ2(eps);

}

void task3()

{

int n, k = 0;

cout << "Программа печатает n членов исключая k член\nВведите n = ";

cin >> n;

cout << "Введите k = ";

cin >> k;

print(n, k);

}

void task4()

{

double eps;

cout << "Программа возвращает номер первого элемента который меньше или равен заданной точности\nВведите eps = ";

cin >> eps;

cout << "Номер = " << findFirstElement(eps);

}

void task5()

{

double eps;

cout << "Программа возвращает номер первого отрицательного элемента который меньше или равен заданной точности\nВведите eps = ";

cin >> eps;

cout << "Номер = " << findFirstNegativeElement(eps);

}

**Результаты работы программ c циклом For**

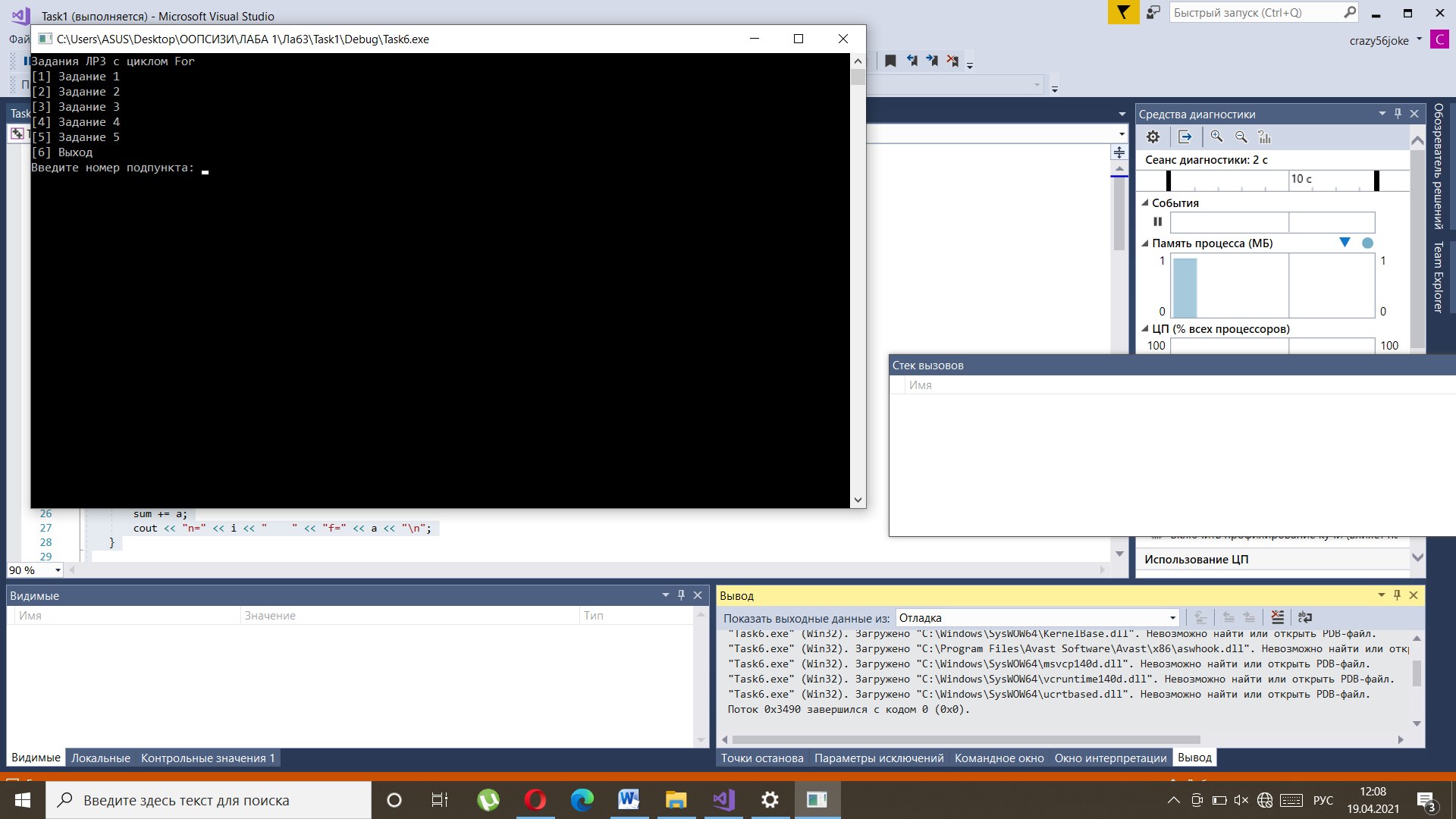


Рисунок 1 –консоль выбора меню

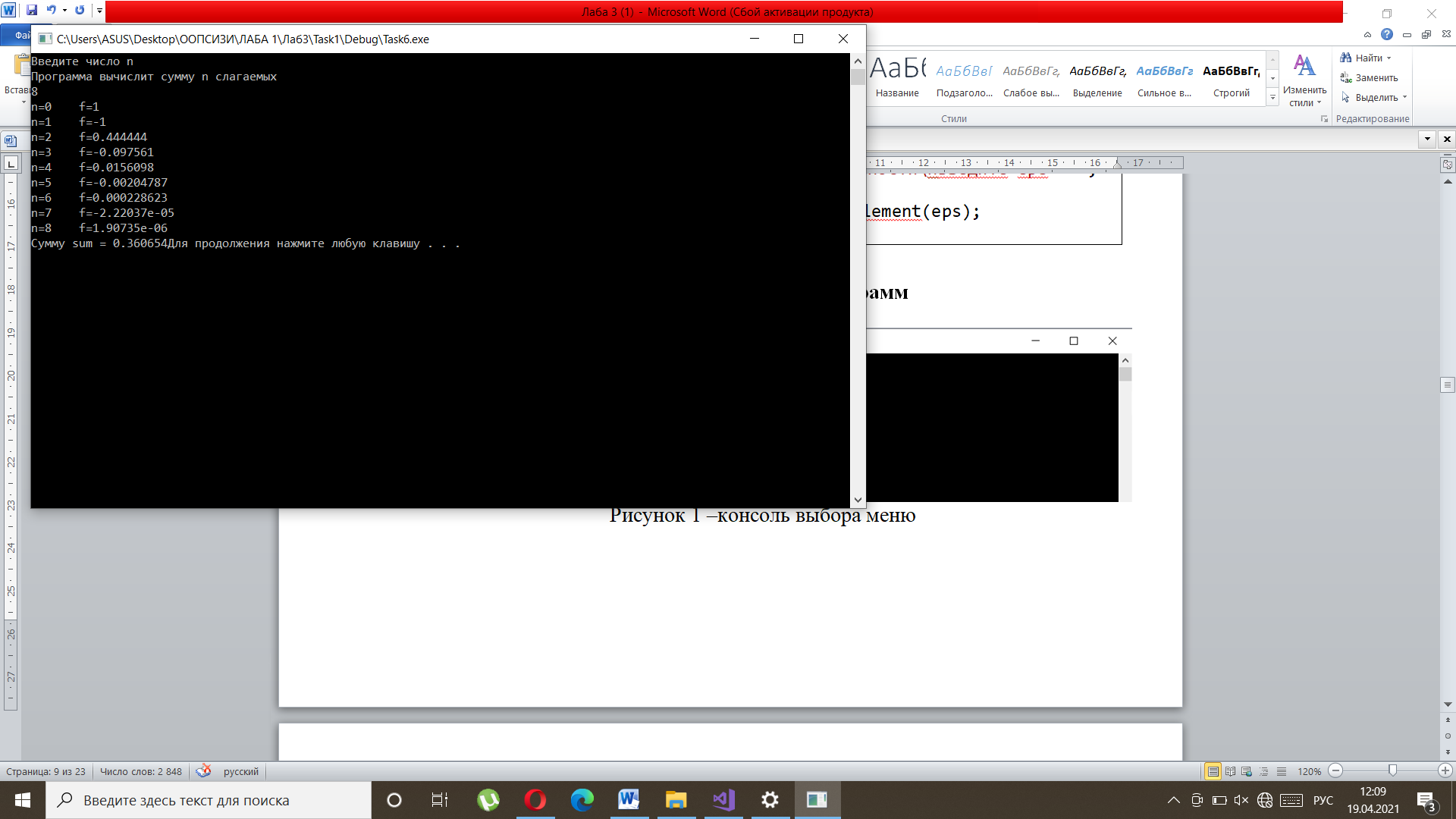


Рисунок 2 – результат выбора задания 1 (цикл For)

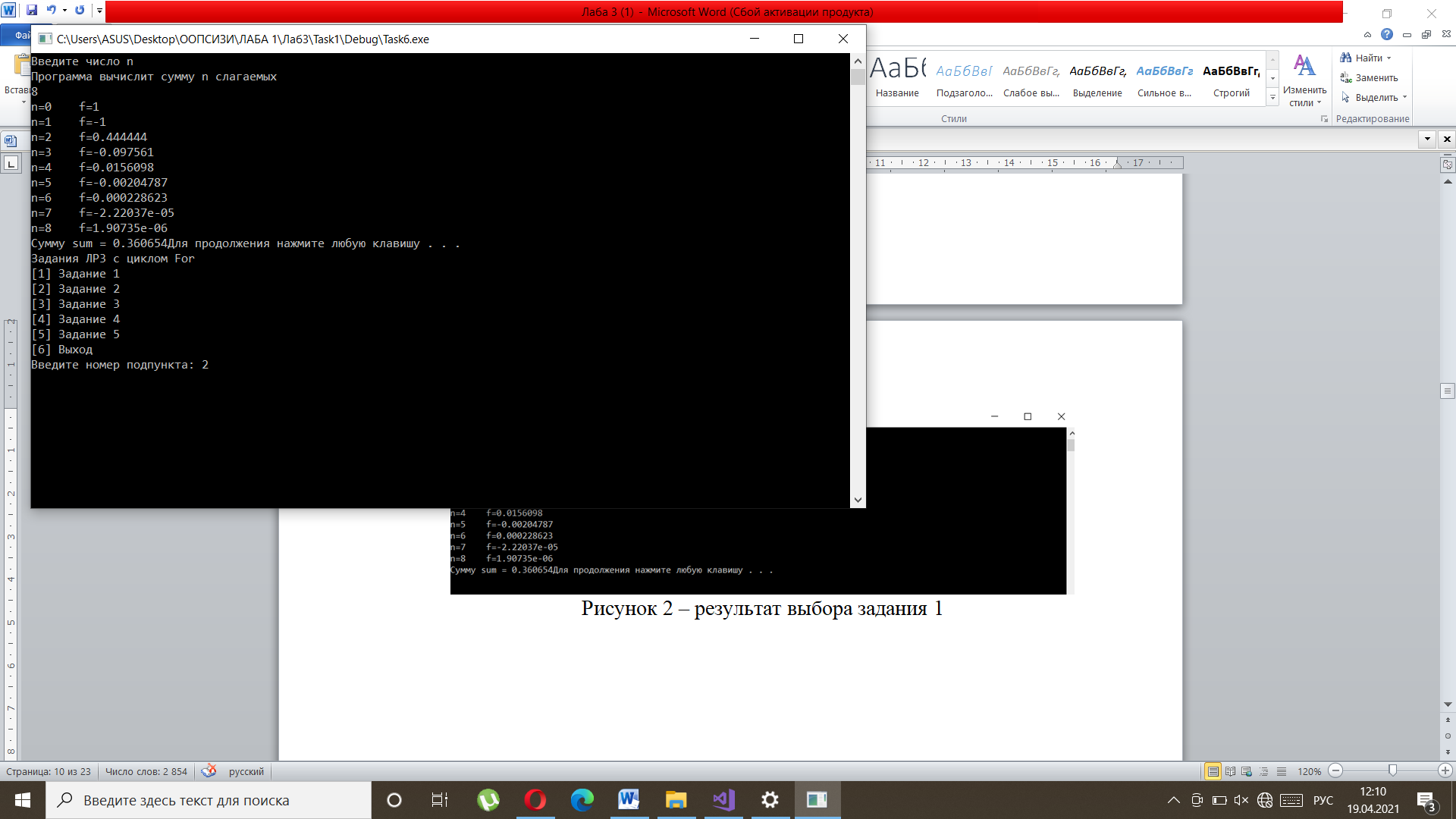


Рисунок 2 – результат перехода обратно в меню и выбора задания 2 (цикл For)

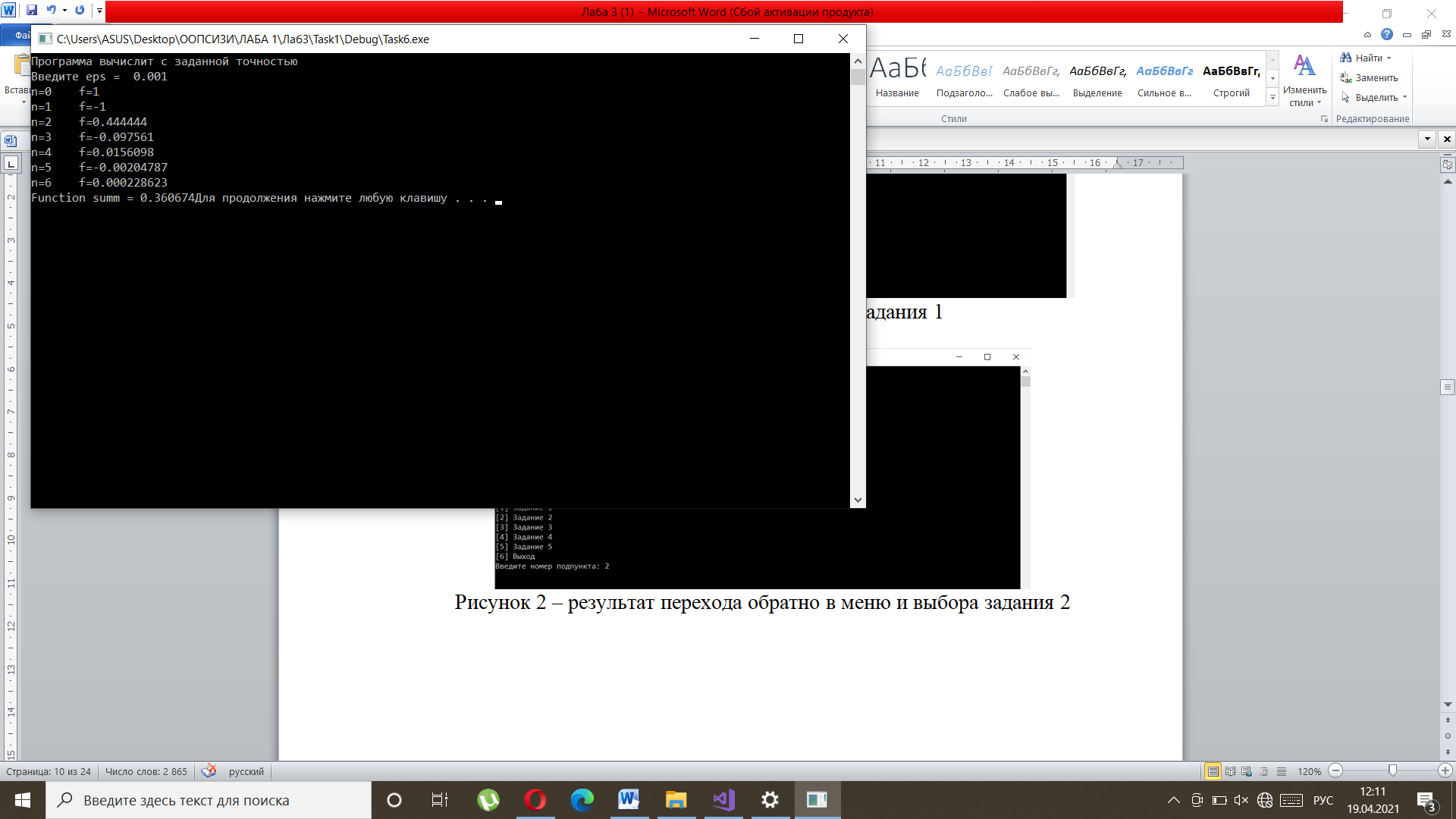


Рисунок 3 – результат задания 2 (цикл For)

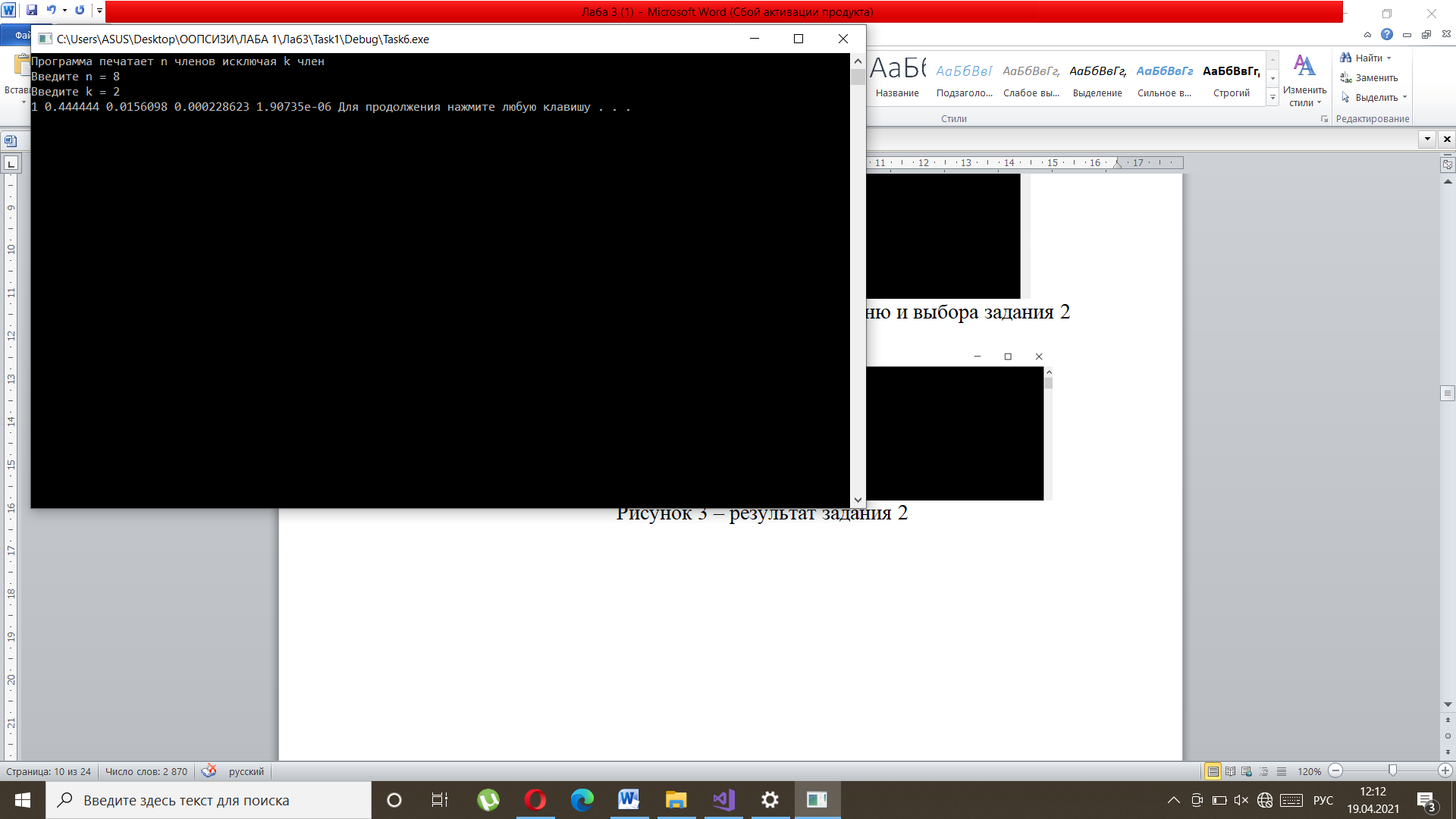


Рисунок 4 – результат задания 3 (цикл For)

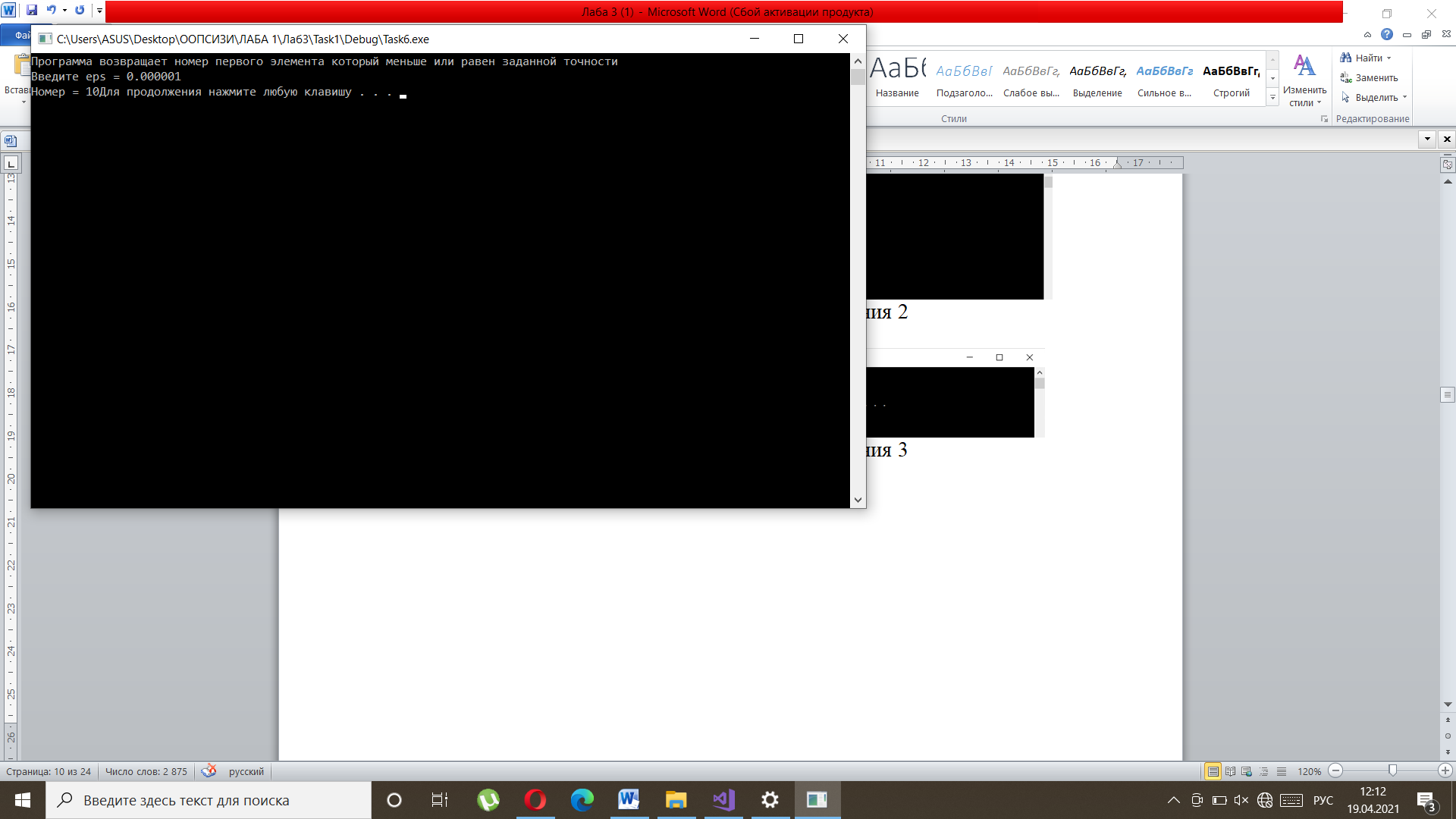


Рисунок 5 – результат задания 4 (цикл For)

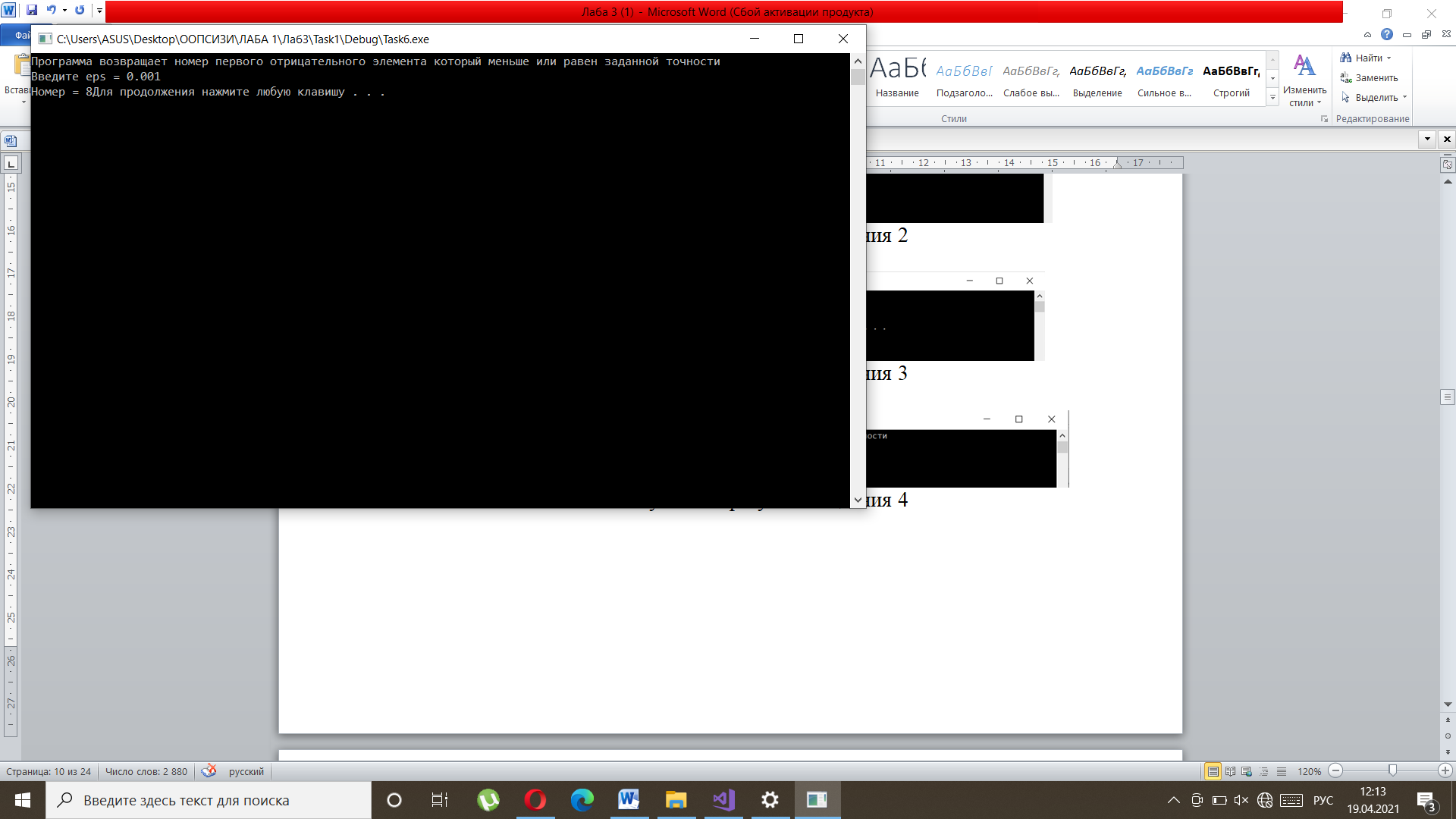


Рисунок 6 – результат задания 5 (цикл For)

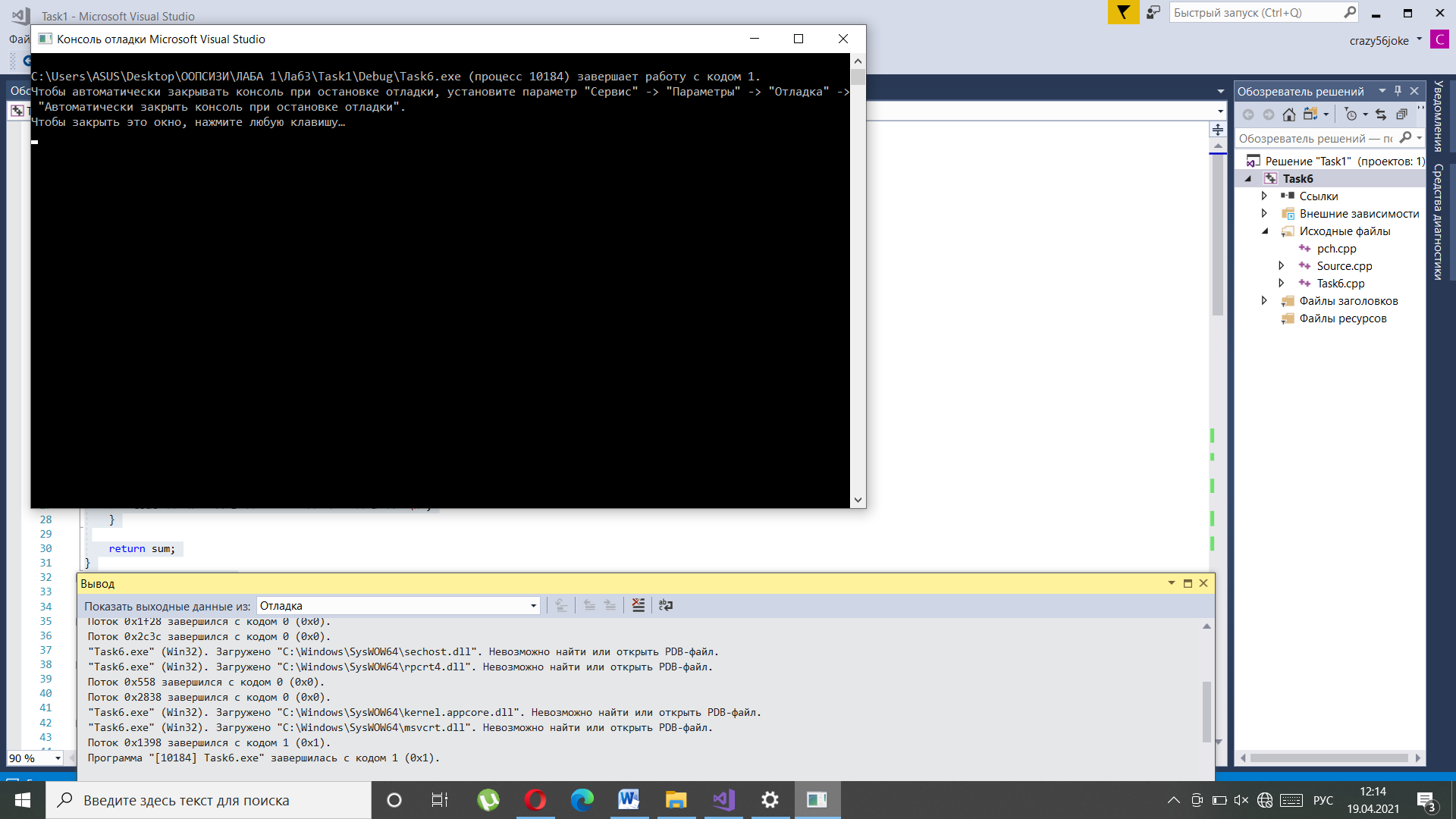


Рисунок 7 – результат задания 6 (цикл For)

**Выполнение с циклом While**

Листинг 8 – прототип функции double summ(int n)

double summ(int n)

{

double a = 0;

double sum = 0;

int i = 0;

while (i <= n)

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

cout << "n=" << i << " " << "f=" << a << "\n";

++i;

}

return sum;

}

Листинг 9 – прототип функции double summ2(double eps)

double summ2(double eps)

{

int i = 0;

double sum = 0;

double a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

while (abs(a) > eps)

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

i++;

}

return sum;

}

Листинг 10 – прототип функции void print(int n, int k)

void print(int n, int k)

{

double count = 0;

int i = 0;

while (i <= n)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if ((i + 1) % k == 0)

{

i++;

continue;

}

else

{

cout << count << " ";

}

i++;

}

}

Листинг 11 – прототип функции int findFirstElement(double eps)

int findFirstElement(double eps)

{

int i = 0;

double count = 0;

while (true)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

i++;

if (abs(count) < eps)

{

break;

}

}

return i+1;

}

Листинг 12 – прототип функции int findFirstNegativeElement(double eps)

int findFirstNegativeElement(double eps)

{

int i = 0;

double count = 0;

while (true)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

i++;

if (abs(count) < eps && count < 0)

{

return i;

}

}

}

Листинг 13 – функция main()

#include "pch.h"

#include <iostream>

using namespace std;

void taskk1(), task2(), task3(), task4(), task5();

void menu()

{

cout << "Задания ЛР3 с циклом For" << endl

<< "[1] Задание 1 " << endl

<< "[2] Задание 2" << endl

<< "[3] Задание 3" << endl

<< "[4] Задание 4" << endl

<< "[5] Задание 5" << endl

<< "[6] Выход" << endl

<< "Введите номер подпункта: ";

}

int main()

{

int g;

setlocale(LC\_CTYPE, "rus");

while (true)

{

menu();

cin >> g;

system("cls");

switch (g)

{

case 1:

taskk1();

break;

case 2:

task2();

break;

case 3:

task3();

break;

case 4:

task4();

break;

case 5:

task5();

break;

case 6:

return true;

}

system("pause");

}

}

Листинг 14 – файл Source.cpp

#include <iostream>

using namespace std;

double summ(int n)

{

double a = 0;

double sum = 0;

int i = 0;

while (i <= n)

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

cout << "n=" << i << " " << "f=" << a << "\n";

++i;

}

return sum;

}

double summ2(double eps)

{

int i = 0;

double sum = 0;

double a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

while (abs(a) > eps)

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

i++;

}

return sum;

}

void print(int n, int k)

{

double count = 0;

int i = 0;

while (i <= n)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if ((i + 1) % k == 0)

{

i++;

continue;

}

else

{

cout << count << " ";

}

i++;

}

}

int findFirstElement(double eps)

{

int i = 0;

double count = 0;

while (true)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

i++;

if (abs(count) < eps)

{

break;

}

}

return i+1;

}

int findFirstNegativeElement(double eps)

{

int i = 0;

double count = 0;

while (true)

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

i++;

if (abs(count) < eps && count < 0)

{

return i;

}

}

}

void taskk1()

{

int i;

cout << "Введите число n\nПрограмма вычислит сумму n слагаемых\n";

cin >> i;

cout << "Сумму sum = " << summ(i);

}

void task2()

{

double eps;

cout << "Программа вычислит с заданной точностью \nВведите eps = ";

cin >> eps;

cout << "Function summ = " << summ2(eps);

}

void task3()

{

int n, k = 0;

cout << "Программа печатает n членов исключая k член\nВведите n = ";

cin >> n;

cout << "Введите k = ";

cin >> k;

print(n, k);

}

void task4()

{

double eps;

cout << "Программа возвращает номер первого элемента который меньше или равен заданной точности\nВведите eps = ";

cin >> eps;

cout << "Номер = " << findFirstElement(eps);

}

void task5()

{

double eps;

cout << "Программа возвращает номер первого отрицательного элемента который меньше или равен заданной точности\nВведите eps = ";

cin >> eps;

cout << "Номер = " << findFirstNegativeElement(eps);

}

**Результат работы программ с циклом While**

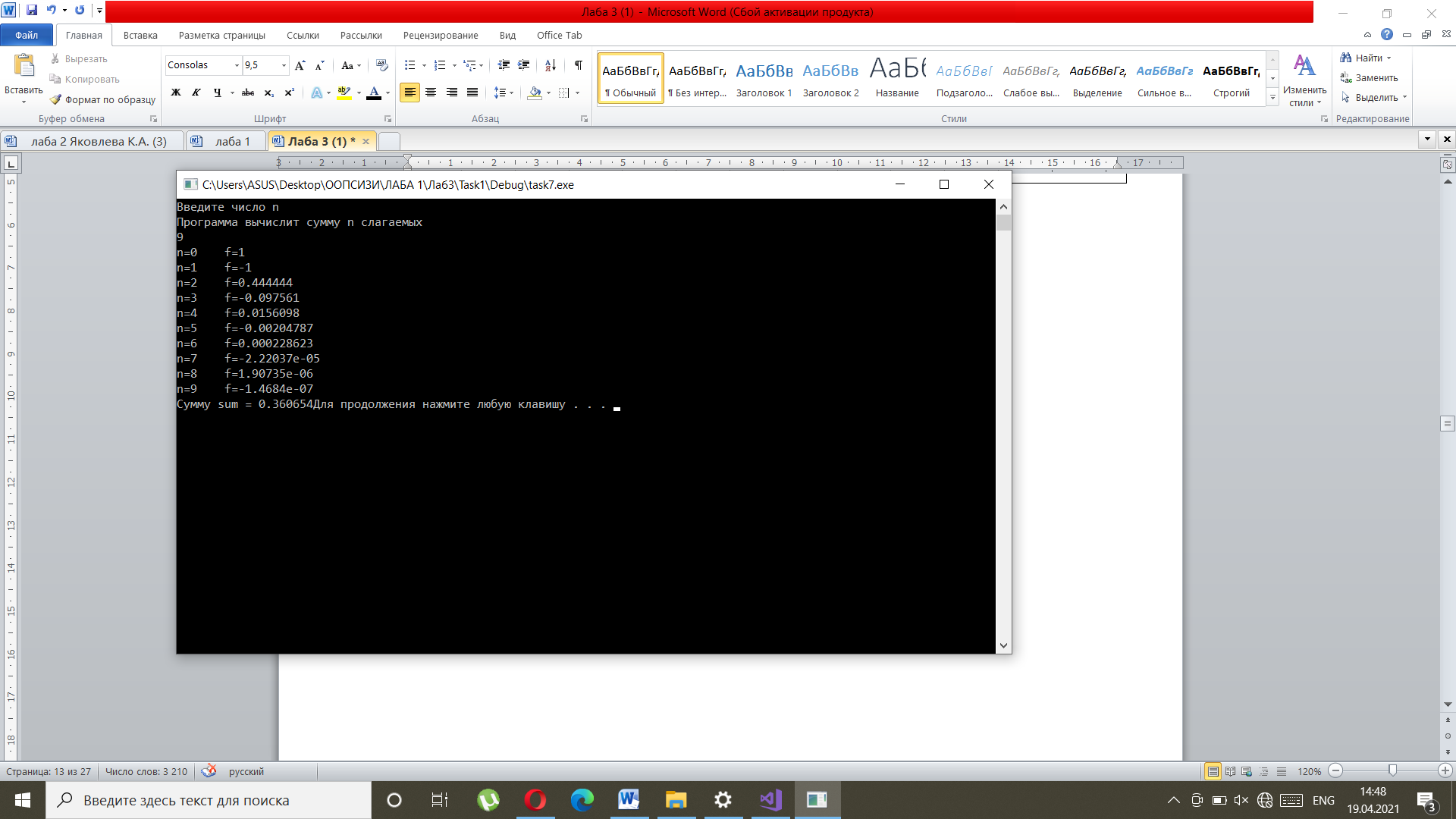


Рисунок 8 – результат выбора задания 1

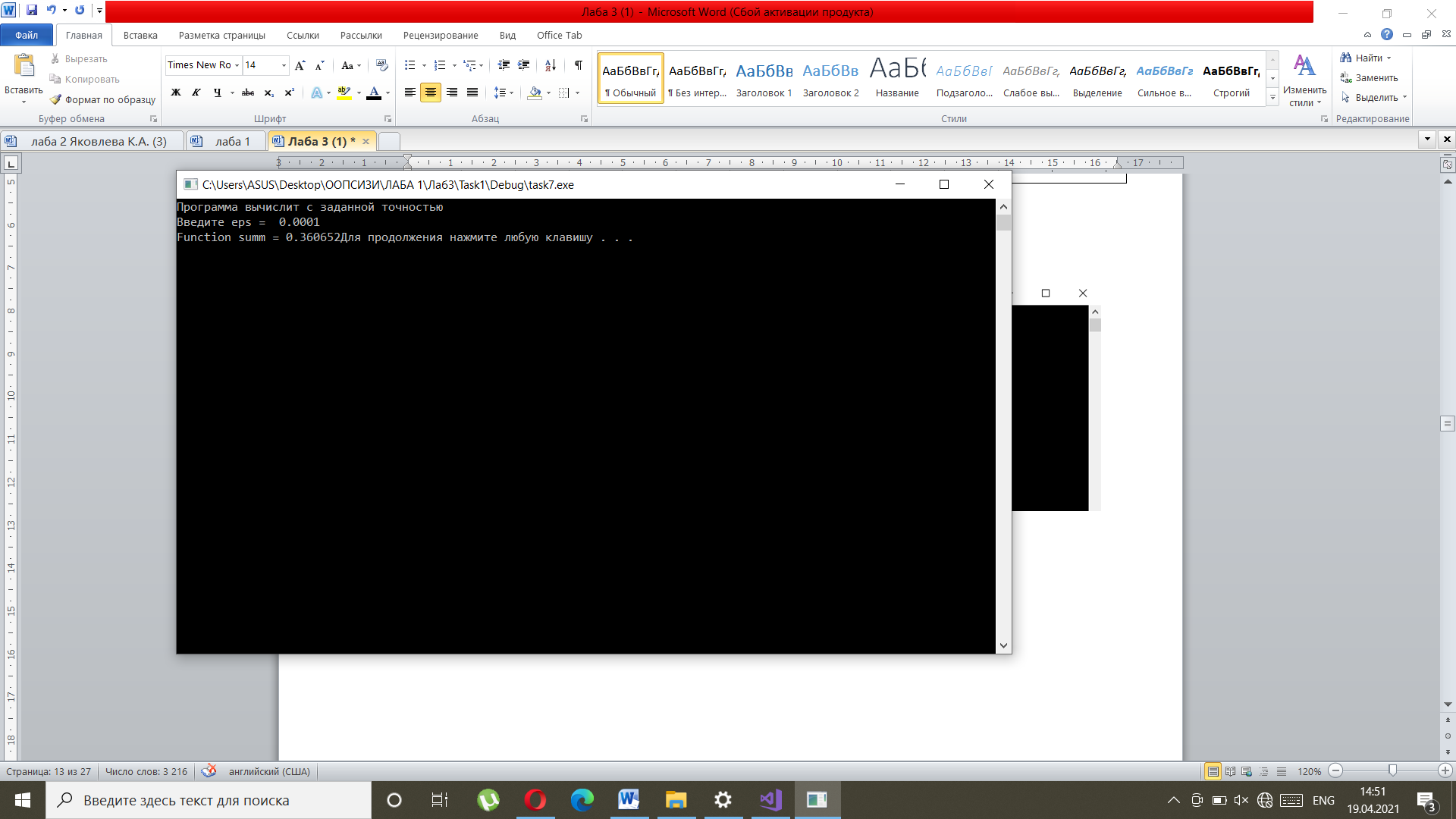


Рисунок 9 – результат выбора задания 2

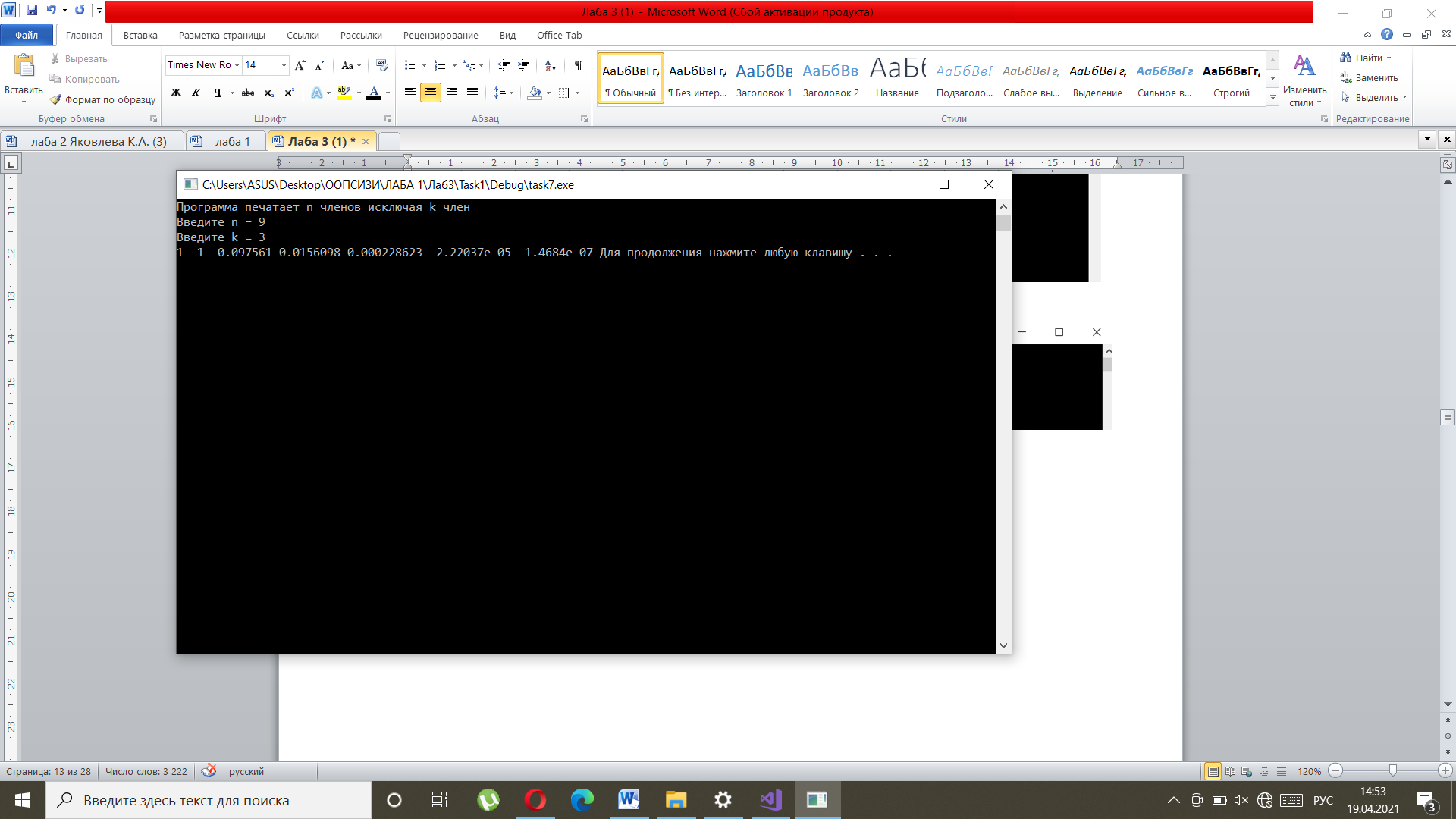


Рисунок 10 – результат выбора задания 3

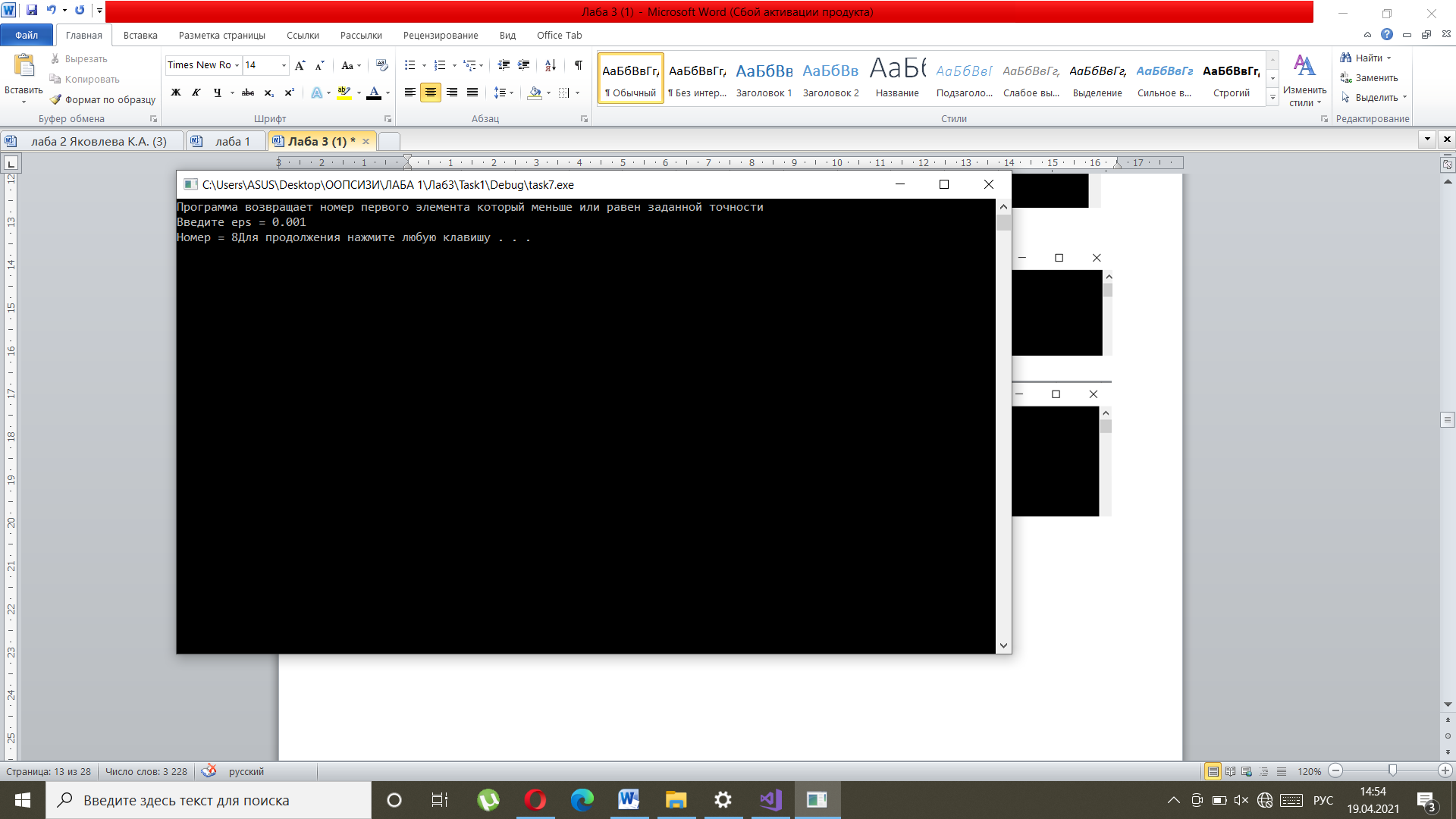


Рисунок 11 – результат выбора задания 4

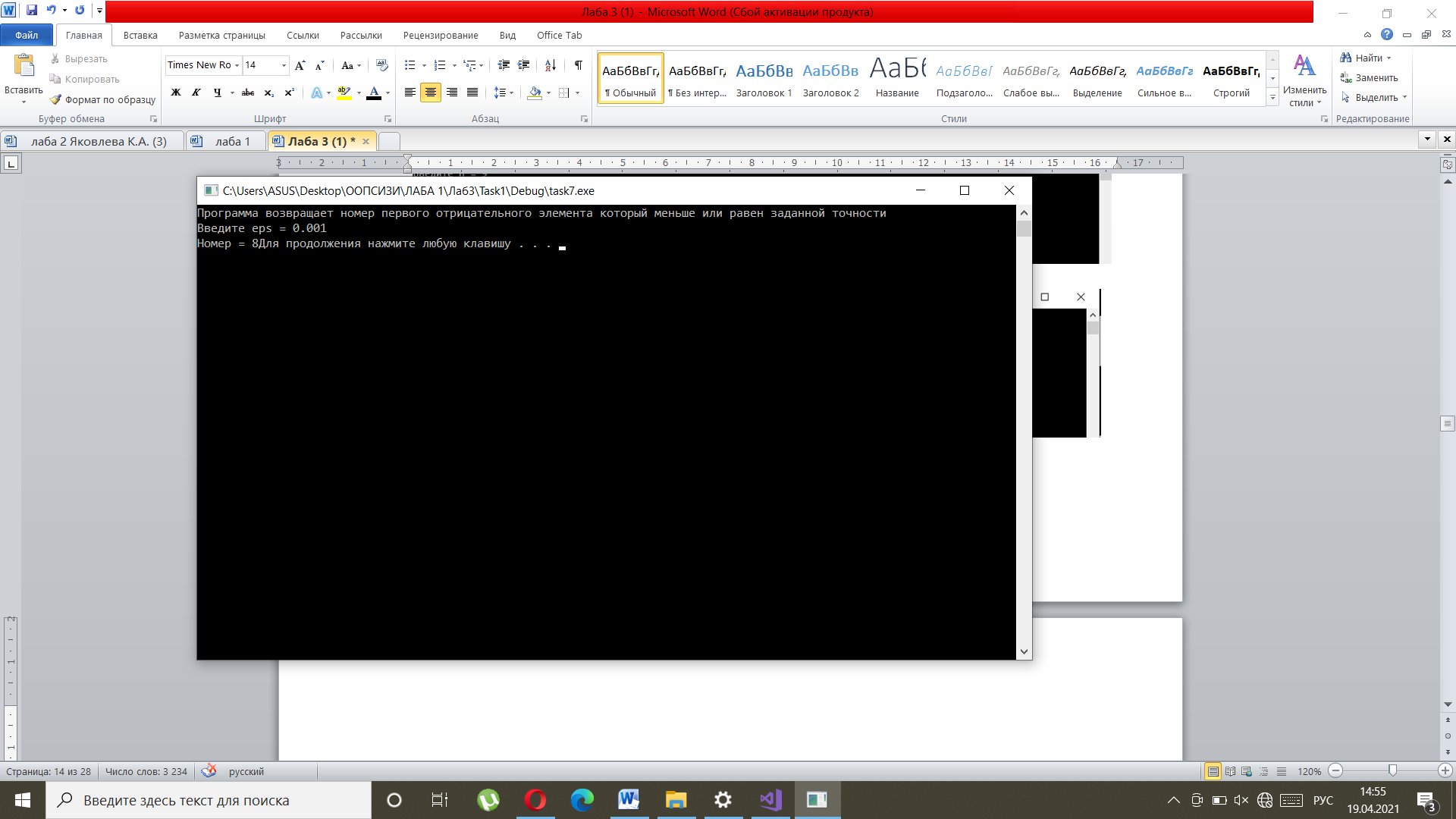


Рисунок 12 – результат выбора задания 5

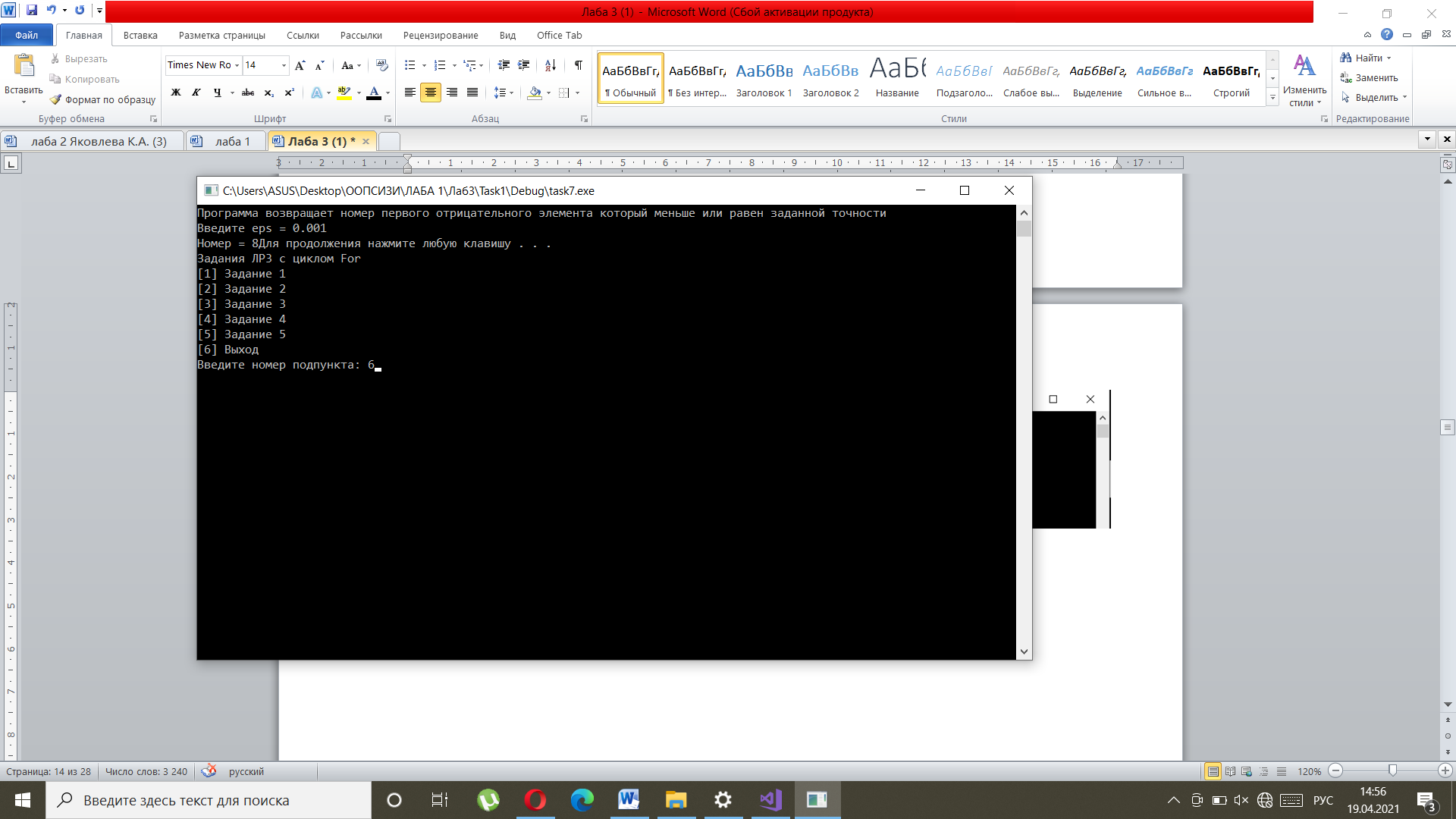


Рисунок 13 – выбор пункта 6

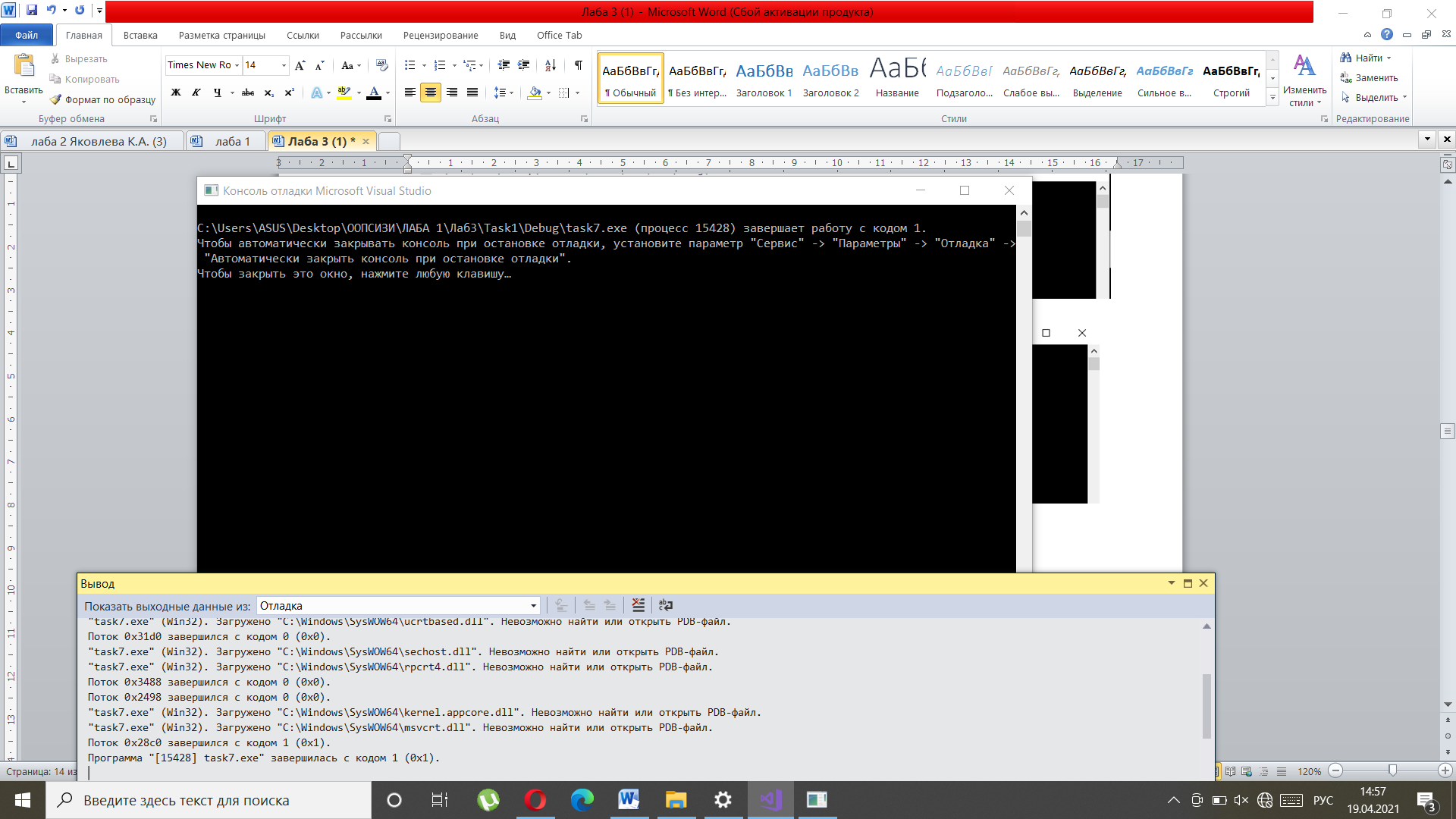


Рисунок 14 – результат выбора пункта 6

**Выполнение с циклом Do..While**

Листинг 15 – прототип функции double summ(int n)

double summ(int n)

{

double a = 0;

double sum = 0;

int i = 0;

do

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

i++;

} while (i <= n);

return sum;

}

Листинг 16 – прототип функции double summ2(double eps)

double summ2(double eps)

{

int i = 0;

double sum = 0;

double a = 0;

do

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

i++;

} while (abs(a) > eps);

return sum;

}

Листинг 17 – прототип функции void print(int n, int k)

void print(int n, int k)

{

double count = 0;

int i = 0;

do

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if ((i + 1) % k == 0)

{

i++;

continue;

}

else

{

cout << count << " ";

}

i++;

} while (i < n);

}

Листинг 18 – прототип функции int findFirstElement(double eps)

int findFirstElement(double eps)

{

int i = 0;

double count = 0;

do

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

i++;

if (abs(count) < eps)

{

break;

}

} while (true);

return i;

}

Листинг 19 – прототип функции int findFirstNegativeElement(double eps)

int findFirstNegativeElement(double eps)

{

int i = 0;

double count = 0;

do

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

i++;

if (abs(count) < eps && count < 0)

{

return i;

}

} while (true);

}

Листинг 20 – функция main()

#include "pch.h"

#include <iostream>

using namespace std;

void taskk1(), task2(), task3(), task4(), task5();

void menu()

{

cout << "Задания ЛР3 с циклом Do..While" << endl

<< "[1] Задание 1 " << endl

<< "[2] Задание 2" << endl

<< "[3] Задание 3" << endl

<< "[4] Задание 4" << endl

<< "[5] Задание 5" << endl

<< "[6] Выход" << endl

<< "Введите номер подпункта: ";

}

int main()

{

int g;

setlocale(LC\_CTYPE, "rus");

while (true)

{

menu();

cin >> g;

system("cls");

switch (g)

{

case 1:

taskk1();

break;

case 2:

task2();

break;

case 3:

task3();

break;

case 4:

task4();

break;

case 5:

task5();

break;

case 6:

return true;

}

system("pause");

}

}

Листинг 21 – файл Source.cpp

#include "pch.h"

#include <iostream>

using namespace std;

double summ(int n)

{

double a = 0;

double sum = 0;

int i = 0;

do

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

i++;

} while (i <= n);

return sum;

}

double summ2(double eps)

{

int i = 0;

double sum = 0;

double a = 0;

do

{

a = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

sum += a;

i++;

} while (abs(a) > eps);

return sum;

}

void print(int n, int k)

{

double count = 0;

int i = 0;

do

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

if ((i + 1) % k == 0)

{

i++;

continue;

}

else

{

cout << count << " ";

}

i++;

} while (i < n);

}

int findFirstElement(double eps)

{

int i = 0;

double count = 0;

do

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

i++;

if (abs(count) < eps)

{

break;

}

} while (true);

return i;

}

int findFirstNegativeElement(double eps)

{

int i = 0;

double count = 0;

do

{

count = pow(-1, i) \* (pow(2, i)) / (pow(i, i + 1) + 1);

i++;

if (abs(count) < eps && count < 0)

{

return i;

}

} while (true);

}

void taskk1()

{

int i;

cout << "Введите число n\nПрограмма вычислит сумму n слагаемых\n";

cin >> i;

cout << "Сумму sum = " << summ(i);

}

void task2()

{

double eps;

cout << "Программа вычислит с заданной точностью \nВведите eps = ";

cin >> eps;

cout << "Function summ = " << summ2(eps);

}

void task3()

{

int n, k = 0;

cout << "Программа печатает n членов исключая k член\nВведите n = ";

cin >> n;

cout << "Введите k = ";

cin >> k;

print(n, k);

}

void task4()

{

double eps;

cout << "Программа возвращает номер первого элемента который меньше или равен заданной точности\nВведите eps = ";

cin >> eps;

cout << "Номер = " << findFirstElement(eps);

}

void task5()

{

double eps;

cout << "Программа возвращает номер первого отрицательного элемента который меньше или равен заданной точности\nВведите eps = ";

cin >> eps;

cout << "Номер = " << findFirstNegativeElement(eps);

}

**Результат выполнения с циклом Do..While**

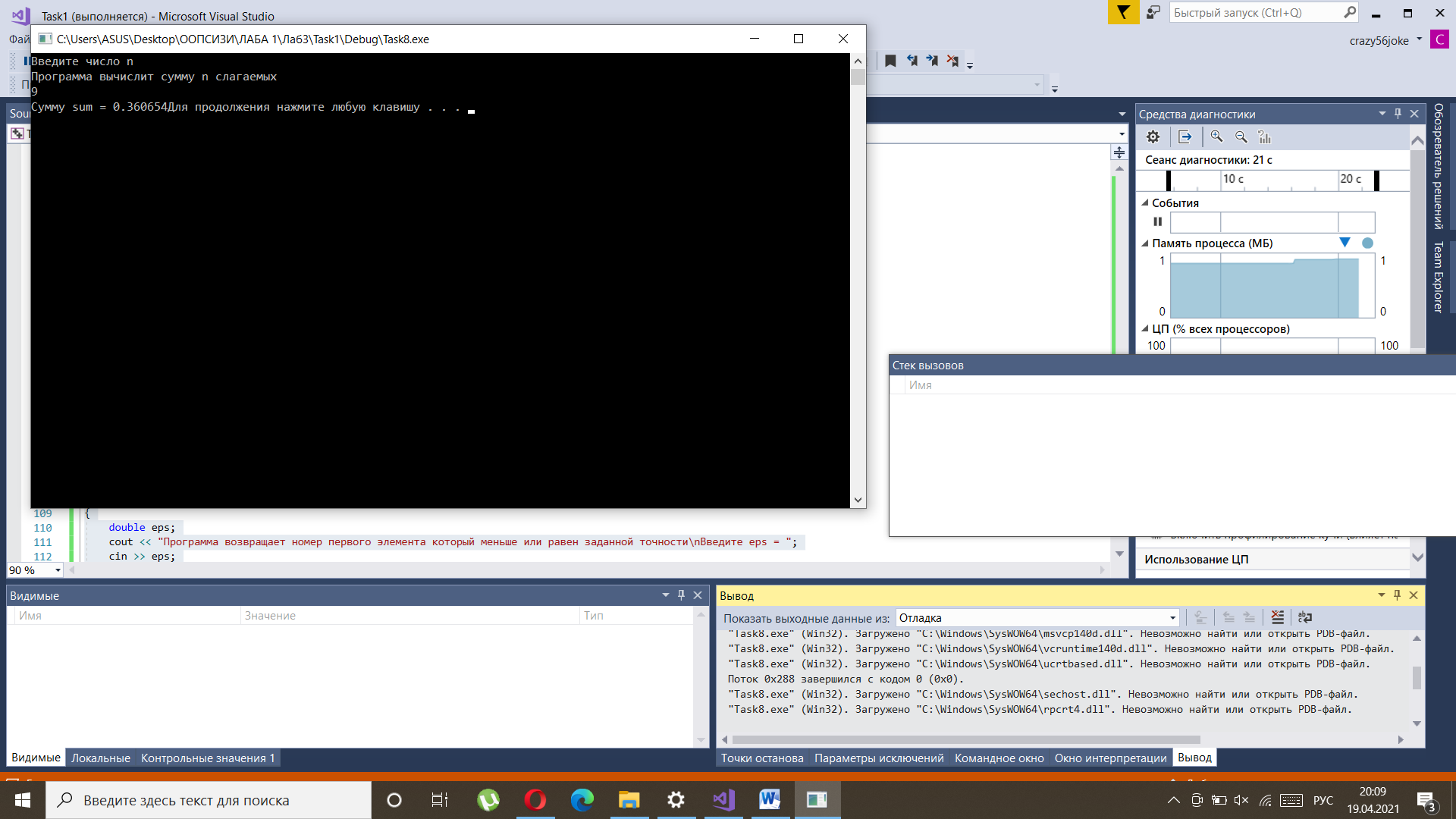


Рисунок 15 – результат выбора задания 1

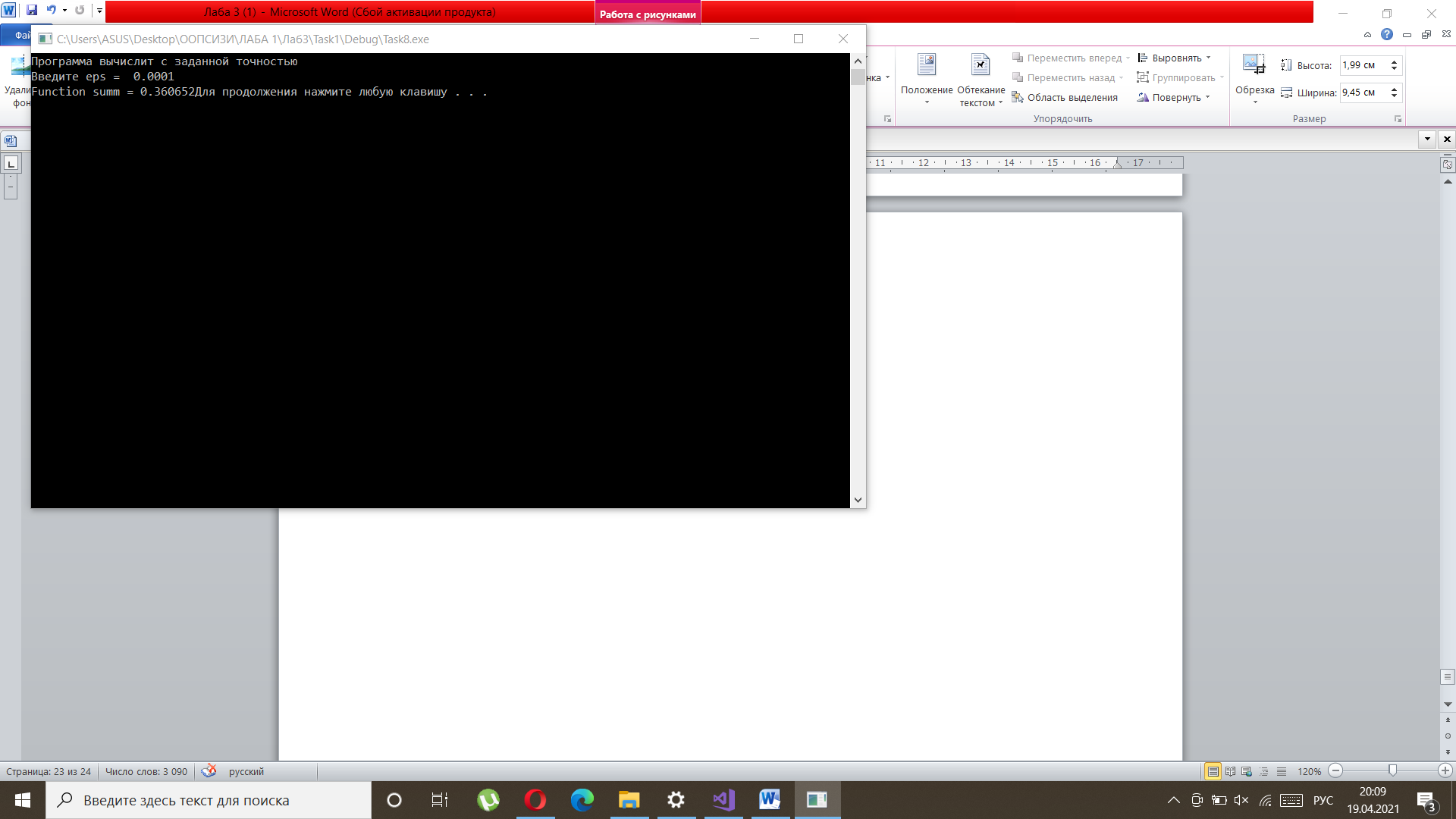


Рисунок 16 – результат выбора задания 2

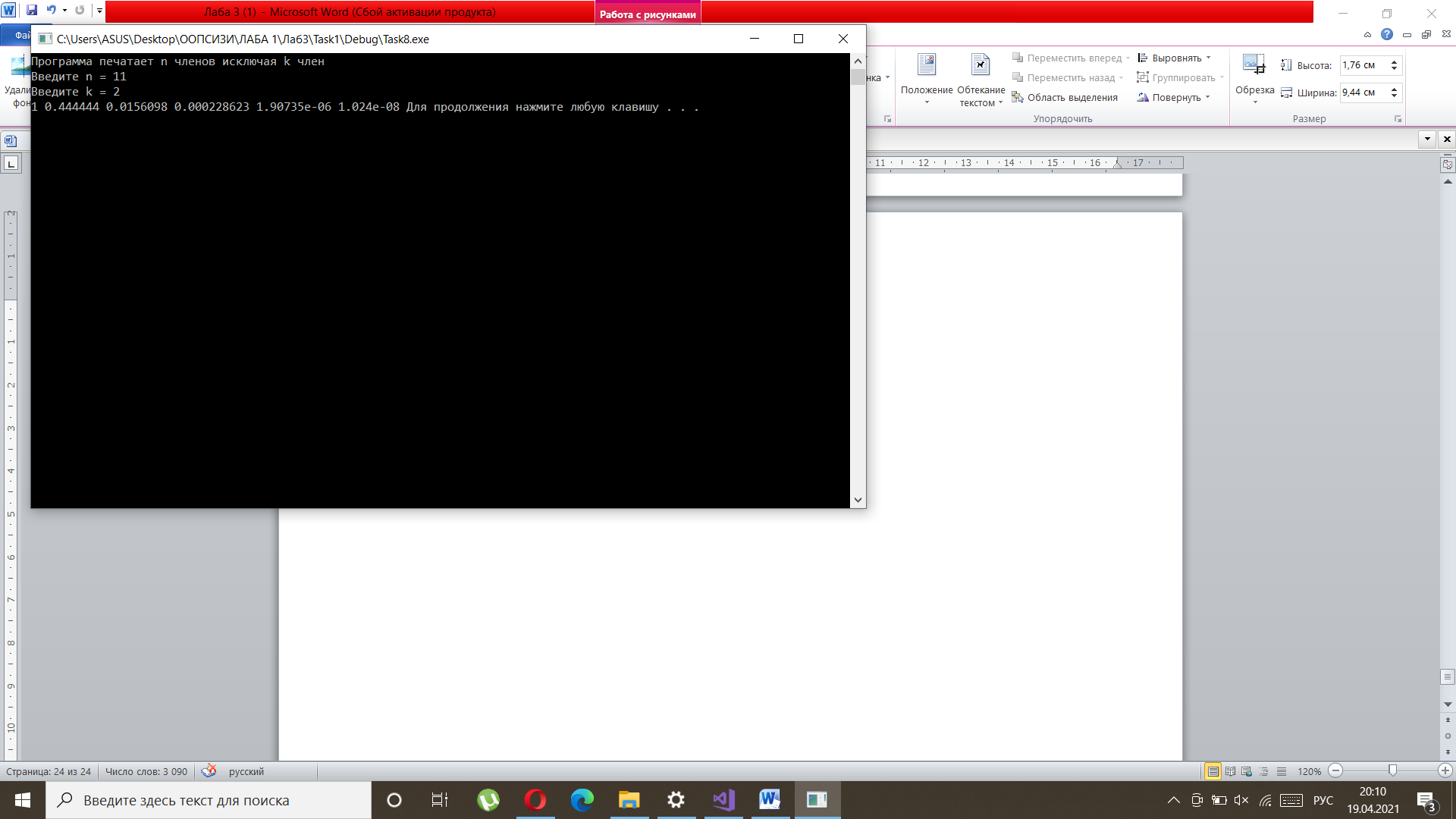


Рисунок 17 – результат выбора задания 3

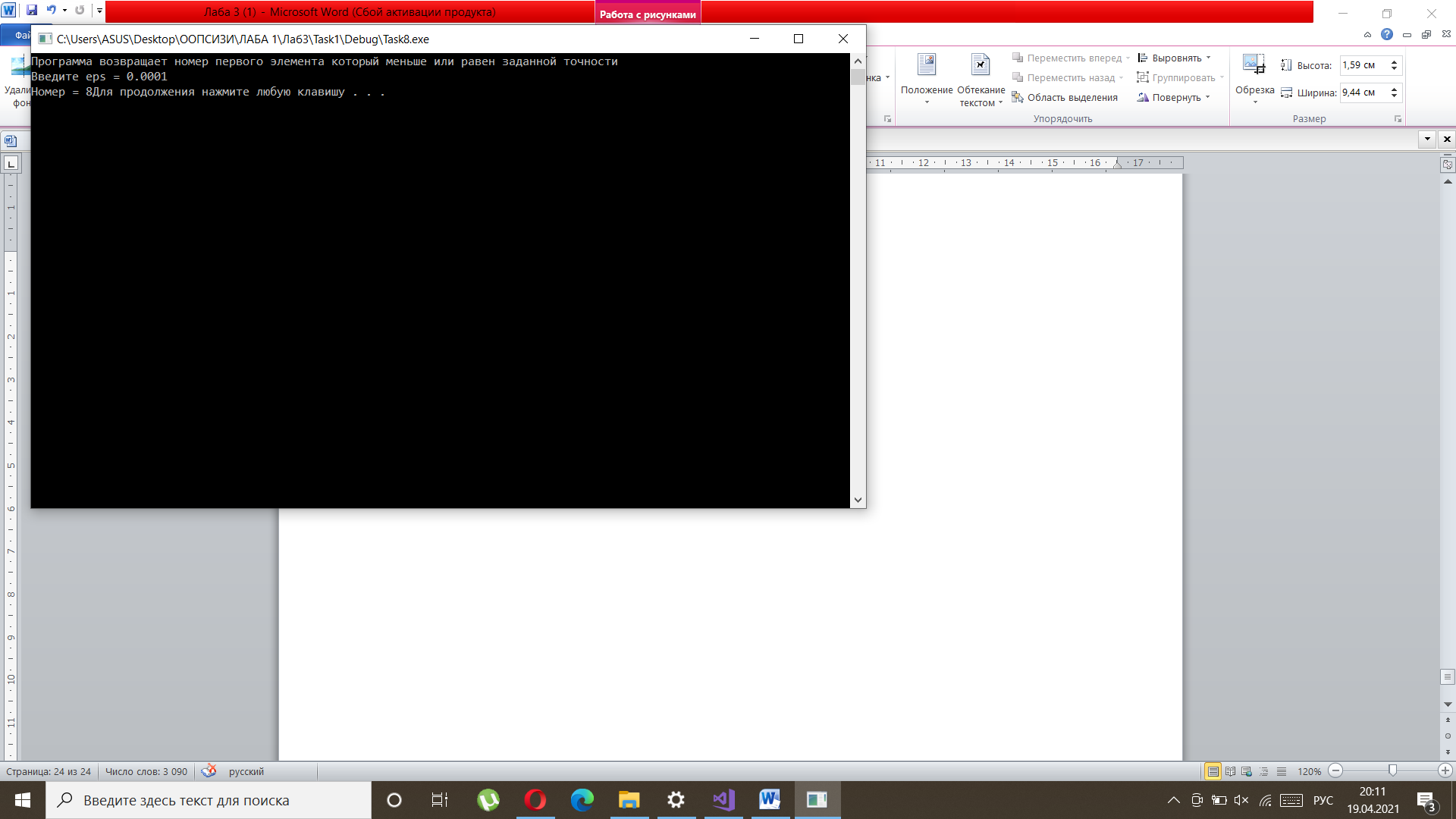


Рисунок 18 – результат выбора задания 4

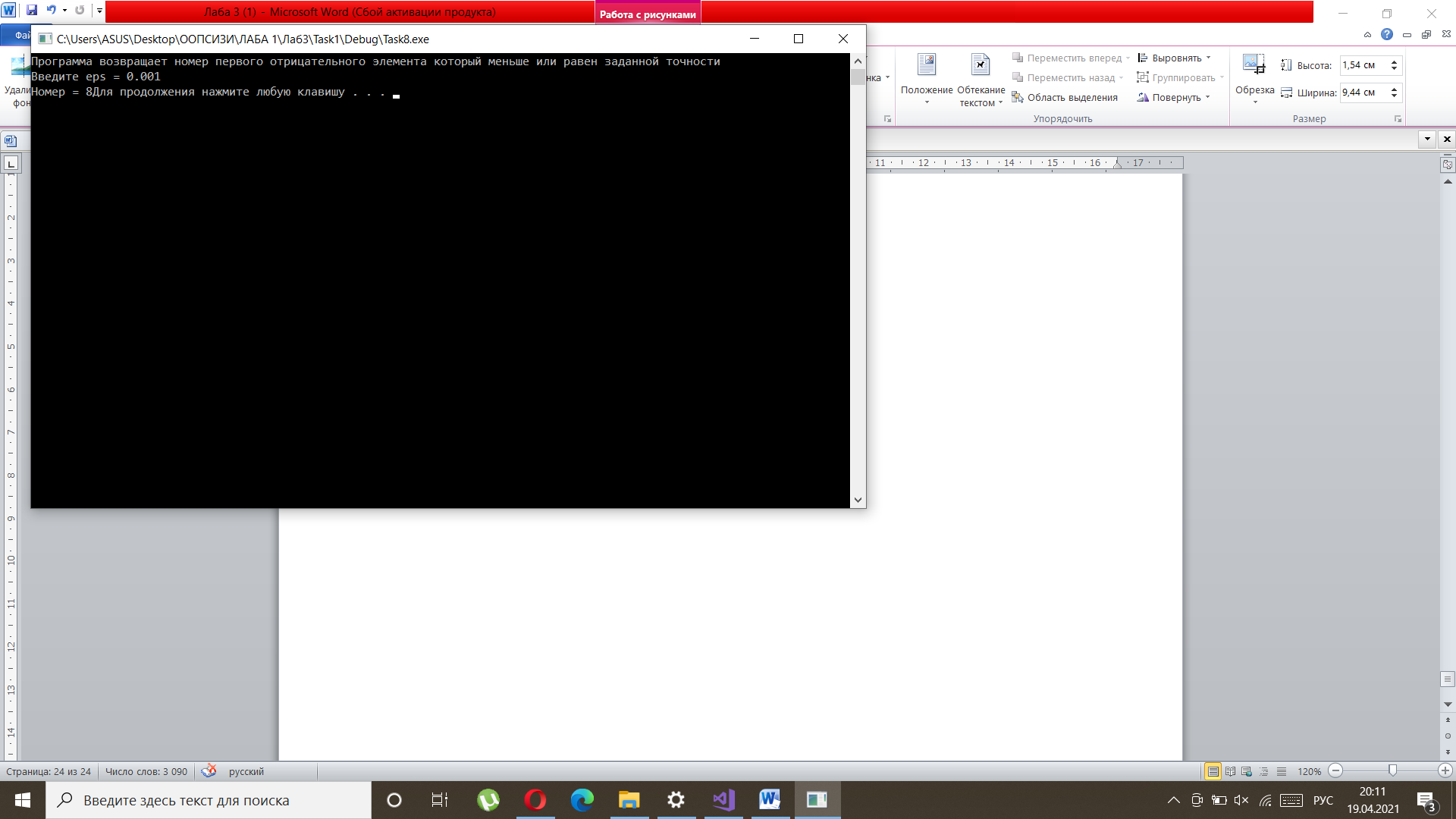


Рисунок 19 – результат выбора задания 5

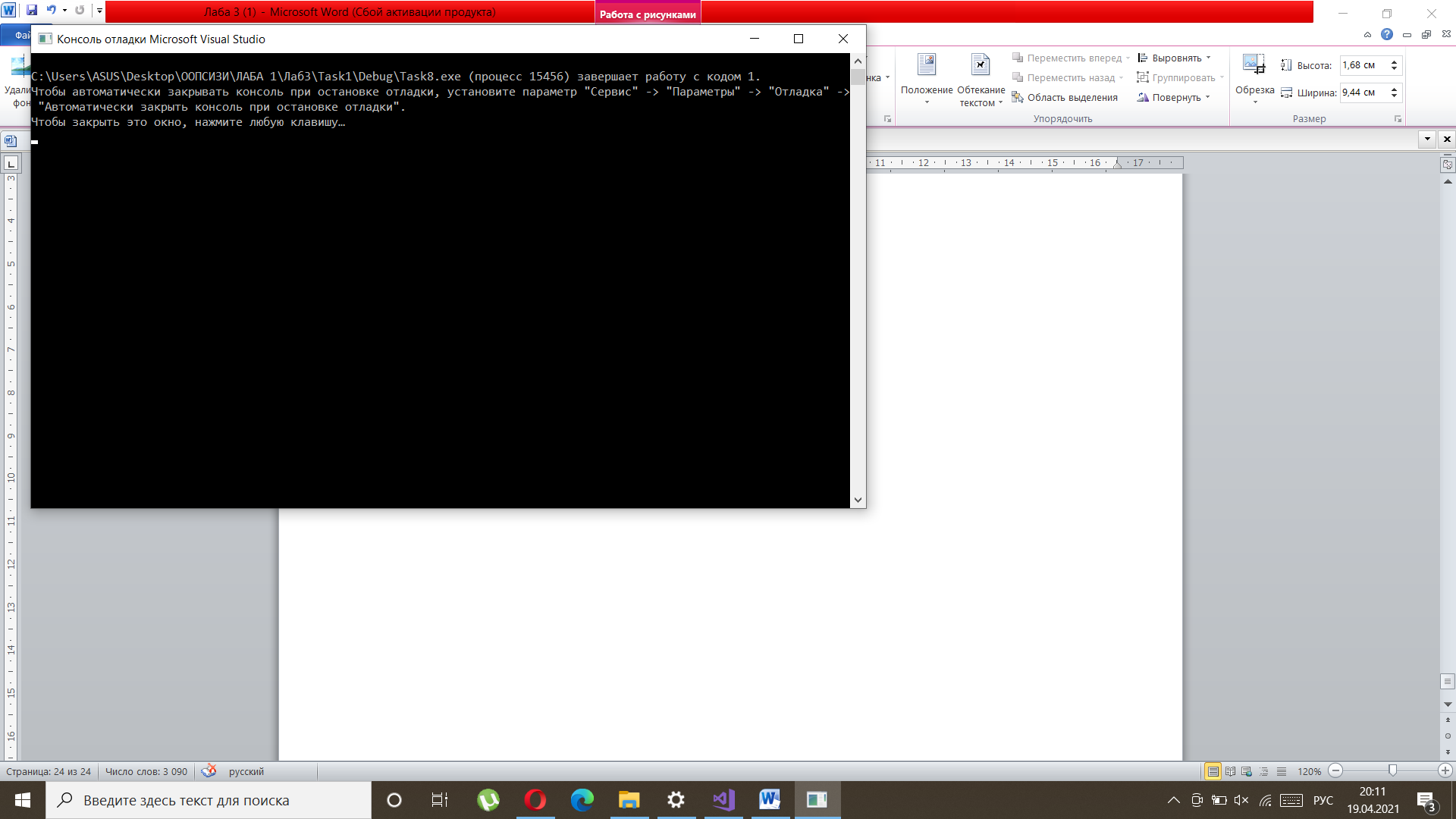


Рисунок 20 – результат выбора задания 6

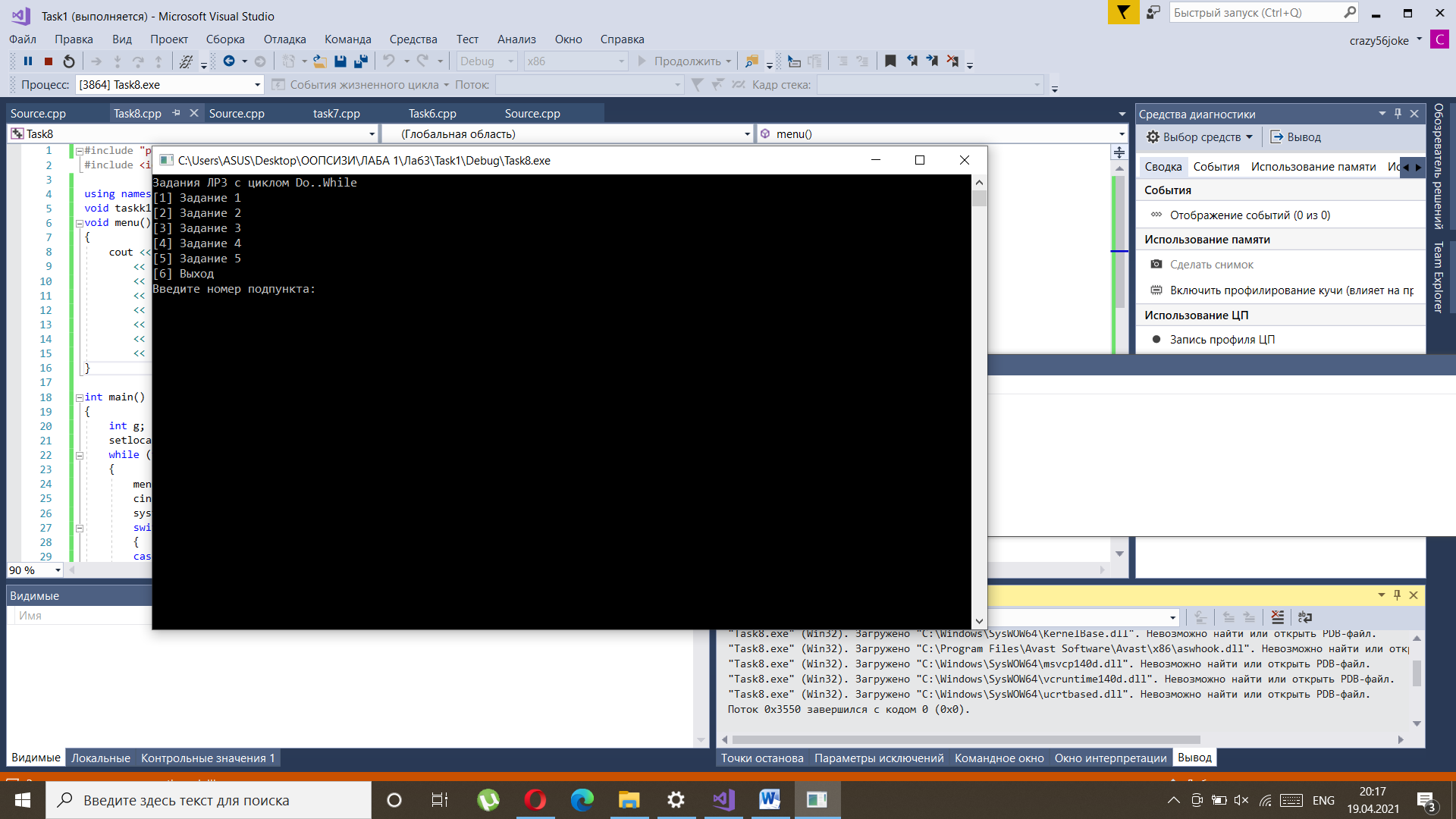


Рисунок 21 – консольное меню задания с циклом Do..While

**Вывод**

В ходе лабораторной работы мною были изучены: циклические алгоритмы на с применением инструкций цикла на примере алгоритмов работы с последовательностями чисел.