ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ассистент |  |  |  | С. Ю. Гуков |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| №3.1.РЕКУРСИВНЫЙ АЛГОРИТМ |
| Вариант 2 (15) |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № |  |  | , |  |  |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

**Оглавление:**

[Цель. 3](#_Toc96548818)

[Индивидуальное задание 4](#_Toc96548819)

[Описание алгоритма 5](#_Toc96548821)

[Блок-схема алгоритма. 6](#_Toc96548822)

[Листинг программы. 10](#_Toc96548823)

[Результат. 16](#_Toc96548824)

[Вывод. 18](#_Toc96548825)

# Цель.

Реализовать программную функцию на языке C/С++, выполняющую поставленную задачу. Вариант задания, пример входных и выходных данных представлен в таблице 7.

* Написать код функции, принимающей в качестве аргументов и возвращающей все необходимые параметры, без использования глобальных переменных. Допустимо использование дополнительных функций.
* Протестировать функцию для всех возможных исключительных ситуаций, особое значение придается текстам на возникновение ошибок в ходе работы программы.
* Из наименования функции и принимаемых аргументов должно быть ясно их назначение.
* В ходе тестирования функции при каждом вызове рекурсивной функции необходимо вывести отладочную информацию: порядковый номер вызова рекурсивной функции, значения изменяющегося аргумента и возвращаемого значения, если они присутствуют. Привести глубину рекурсии для каждого тестового примера.

# Индивидуальное задание.

## 

Рисунок 1. Индивидуальное задание

# Описание алгоритма

С клавиатуры вводится число и проверяется на ввод целого числа. Вызывается функция find.

Функция find. В массив levels добавляется значение глубины, в массив memory добавляется значение входящей переменной х. Если deep\_level > max\_deep\_level, тогда max\_deep\_level = deep\_level. Если х не равен 0, тогда выводится количество вызовов, количество единиц, остаток от деления x % 2, deep\_level, значение x. Если x % 2 == 1, то счетчик единиц увеличивается на 1. Дальше возвращает функцию find, но функция принимает x % 2, и увеличивает deep\_level на 1. Если х == 0, тогда выводится количество вызовов, количество единиц, пустая строка, deep\_level, значение x.

# Блок-схема алгоритма.

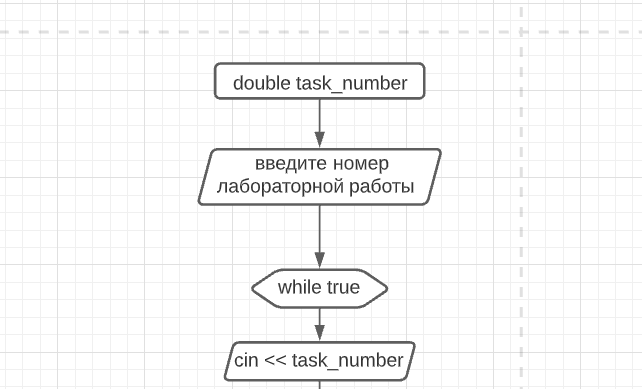


Рисунок 2. Блок схема алгоритма

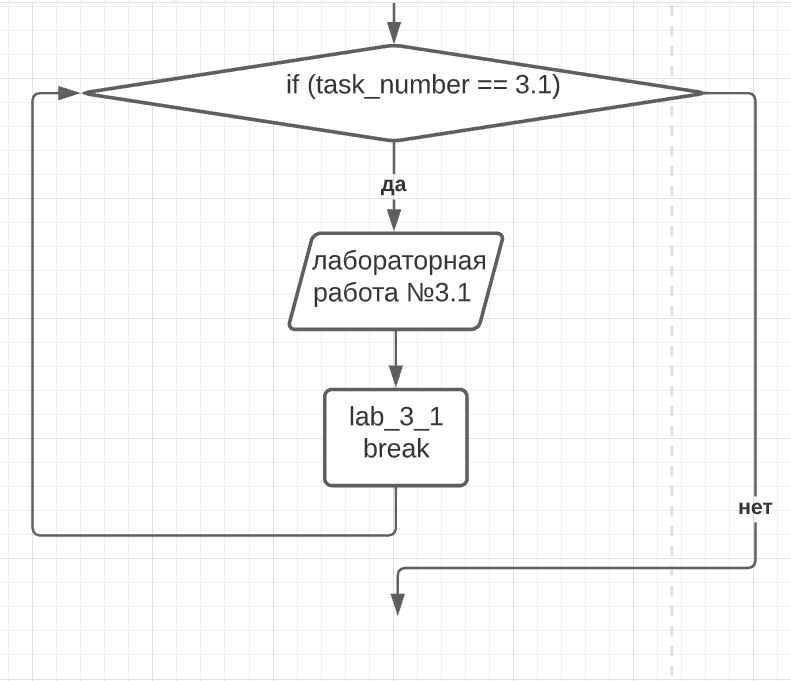


Рисунок 3. Блок схема алгоритма

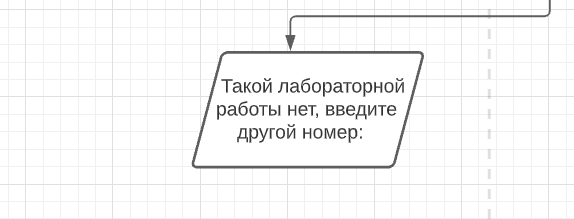


Рисунок 4. Блок схема алгоритма

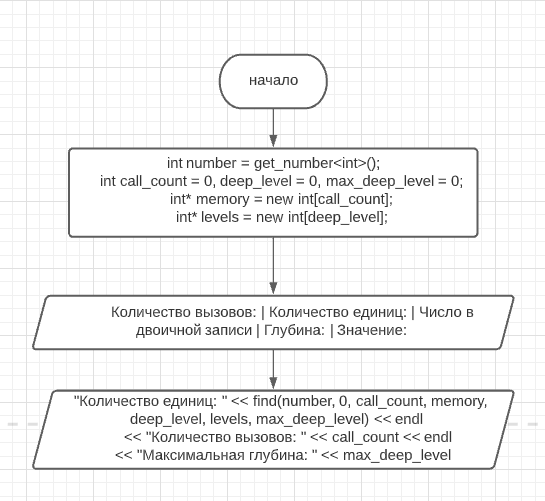


Рисунок 5. Блок схема алгоритма

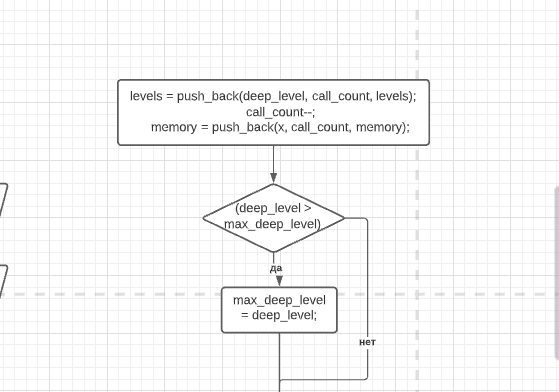


Рисунок 6. Блок схема алгоритма

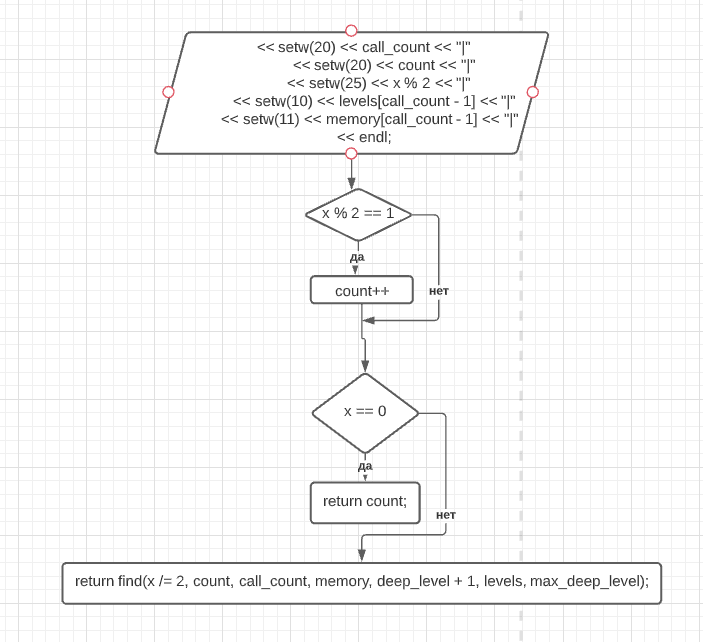


Рисунок 7. Блок схема алгоритма

# Листинг программы.

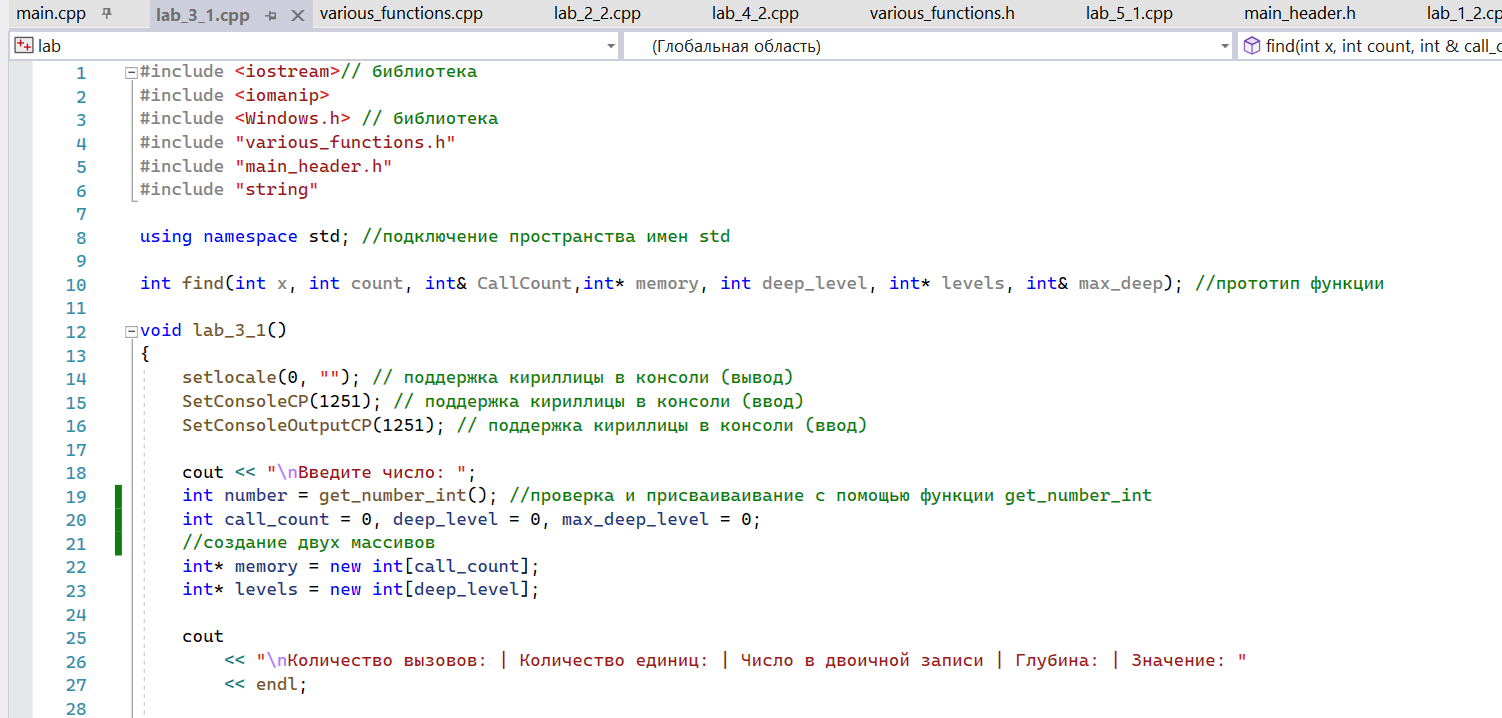


Рисунок 8. Листинг программы lab\_3\_1.cpp

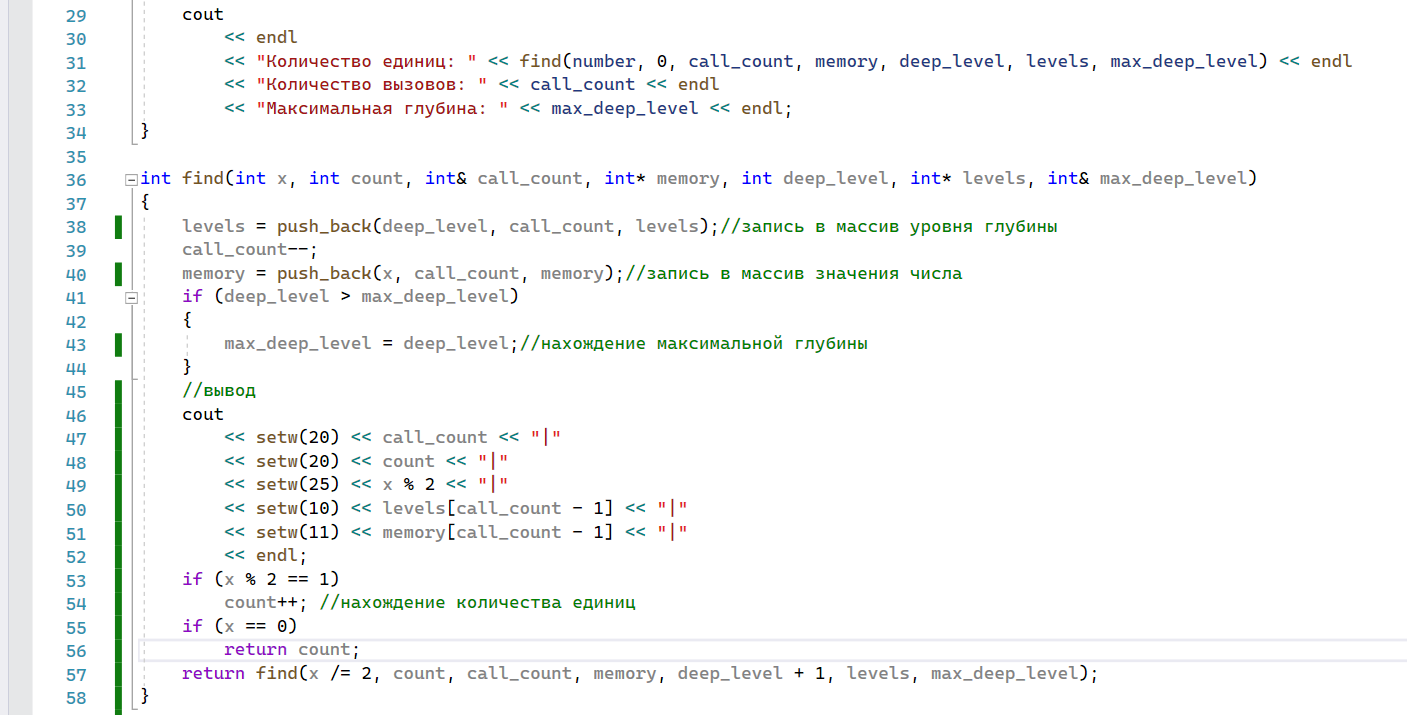


Рисунок 9. Листинг программы lab\_3\_1.cpp

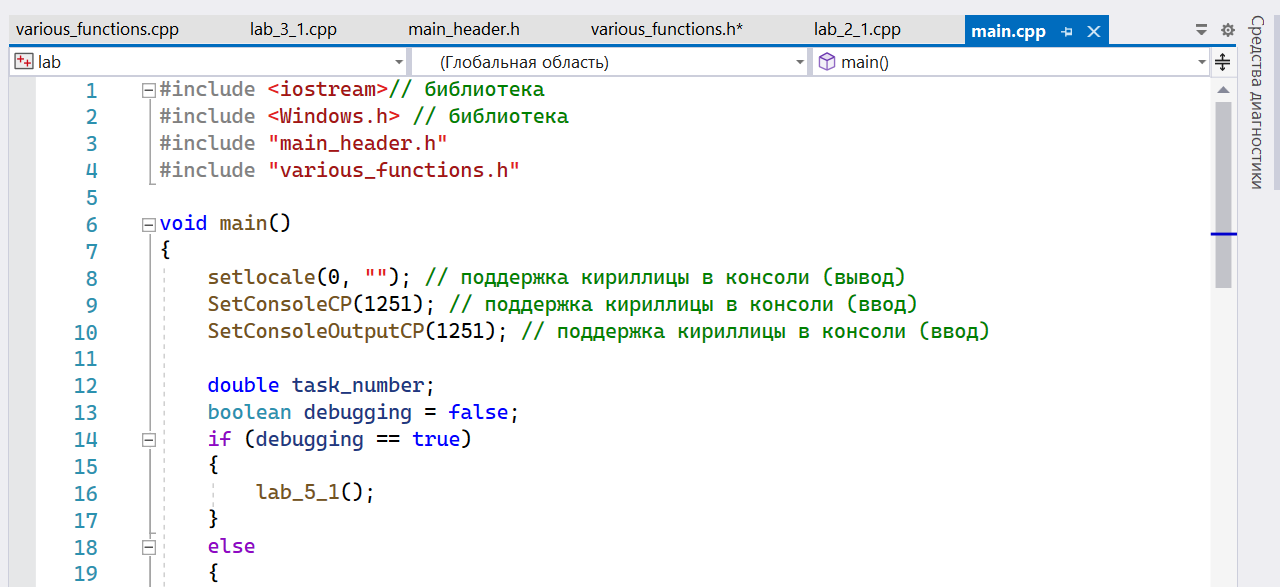


Рисунок 10. Листинг программы main.cpp

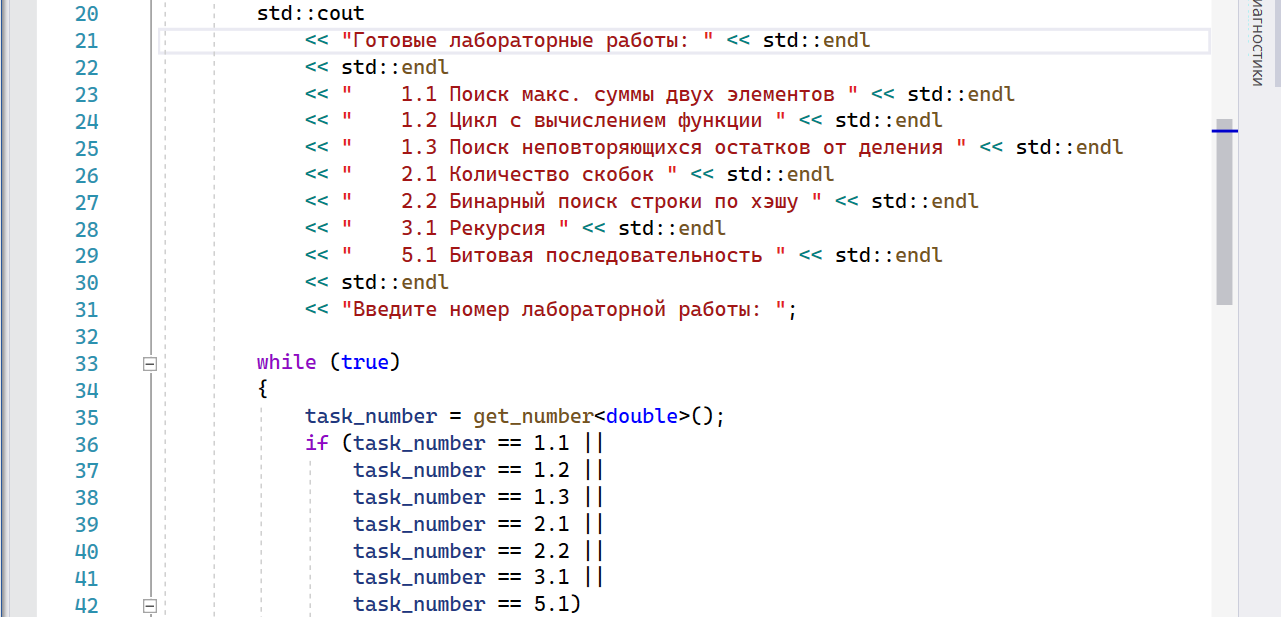


Рисунок 11. Листинг программы main.cpp

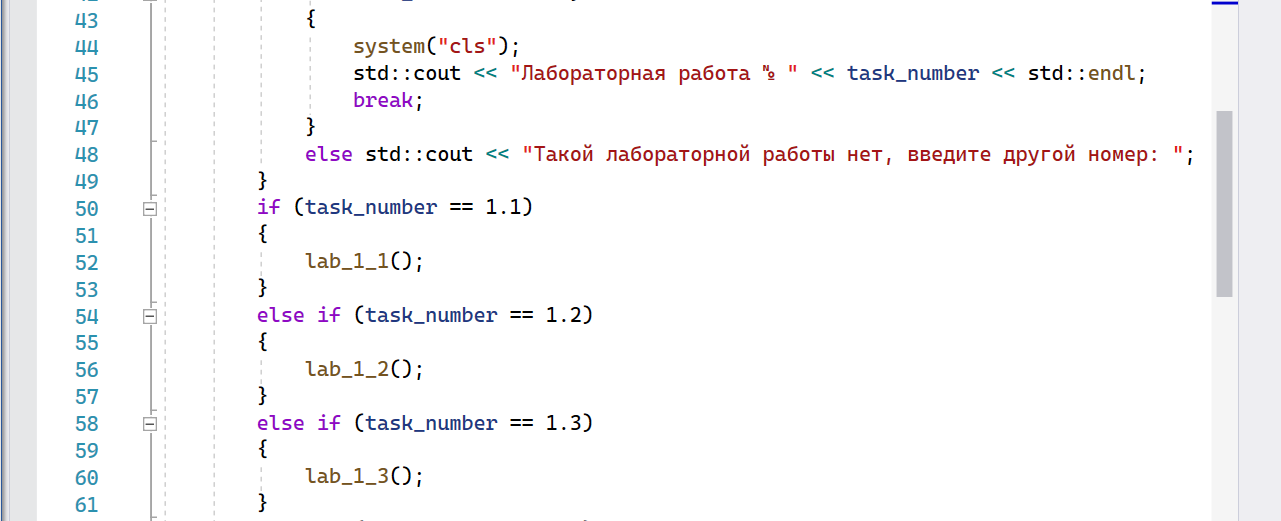


Рисунок 12. Листинг программы main.cpp

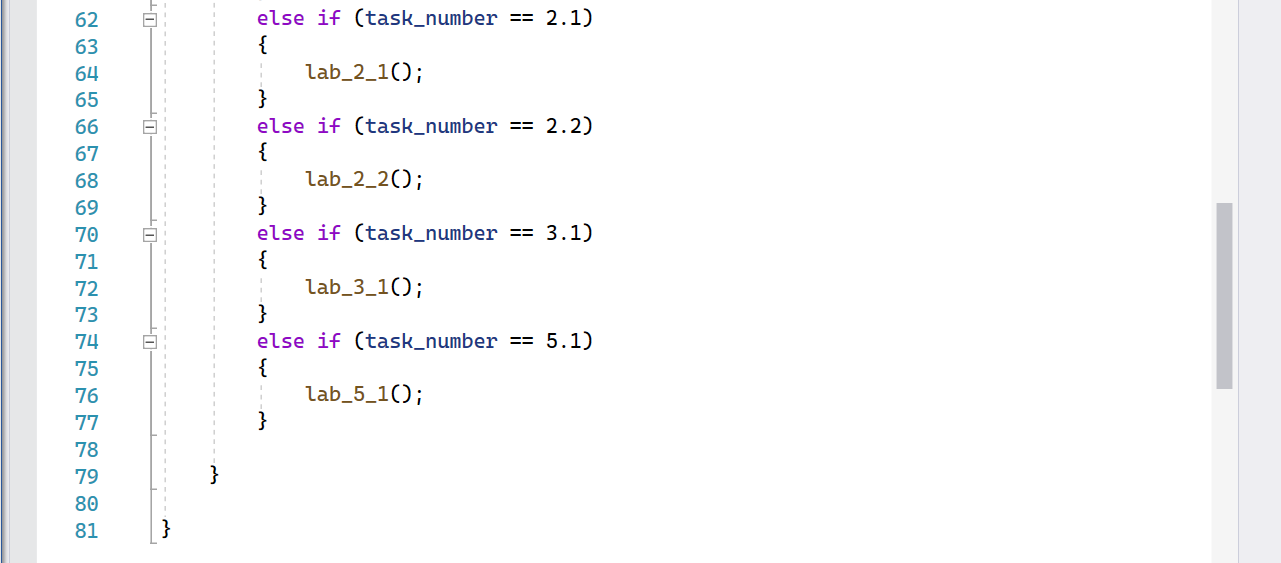


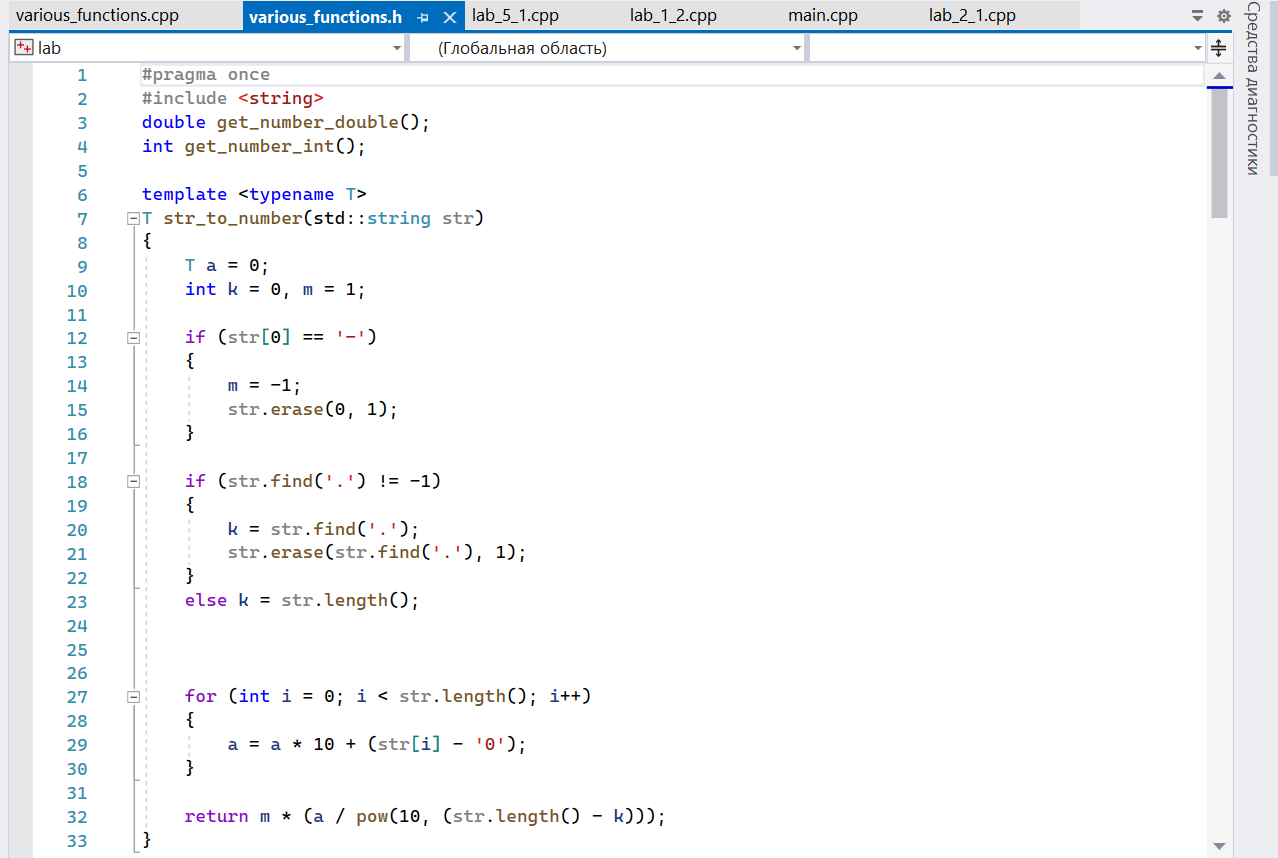
Рисунок 13. Листинг программы main.cpp

Рисунок 14. Листинг программы various\_functions.h и various\_functions.cpp

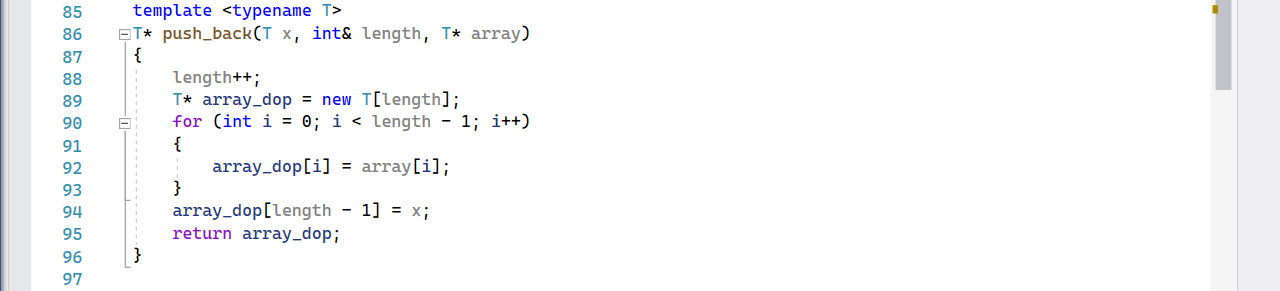


Рисунок 15. Листинг программы various\_functions.h и various\_functions.cpp

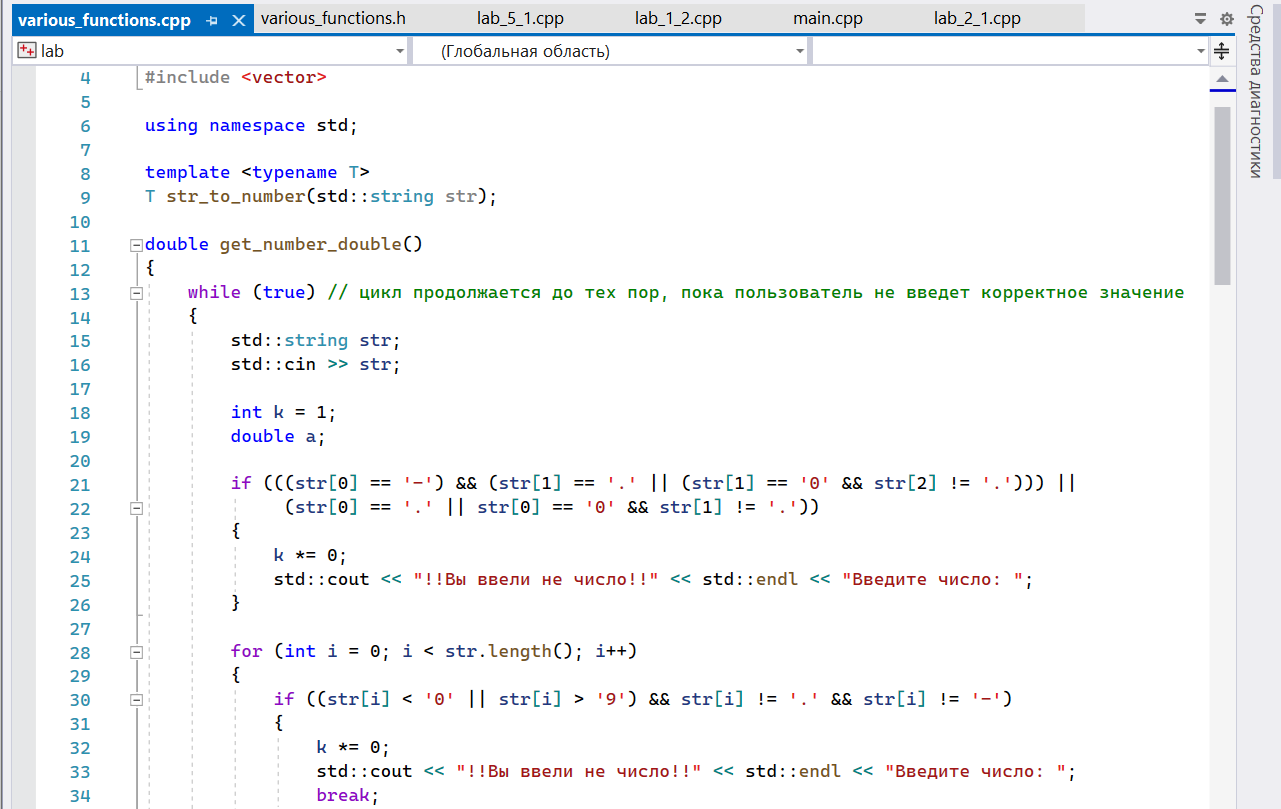


Рисунок 16. Листинг программы various\_functions.h и various\_functions.cpp

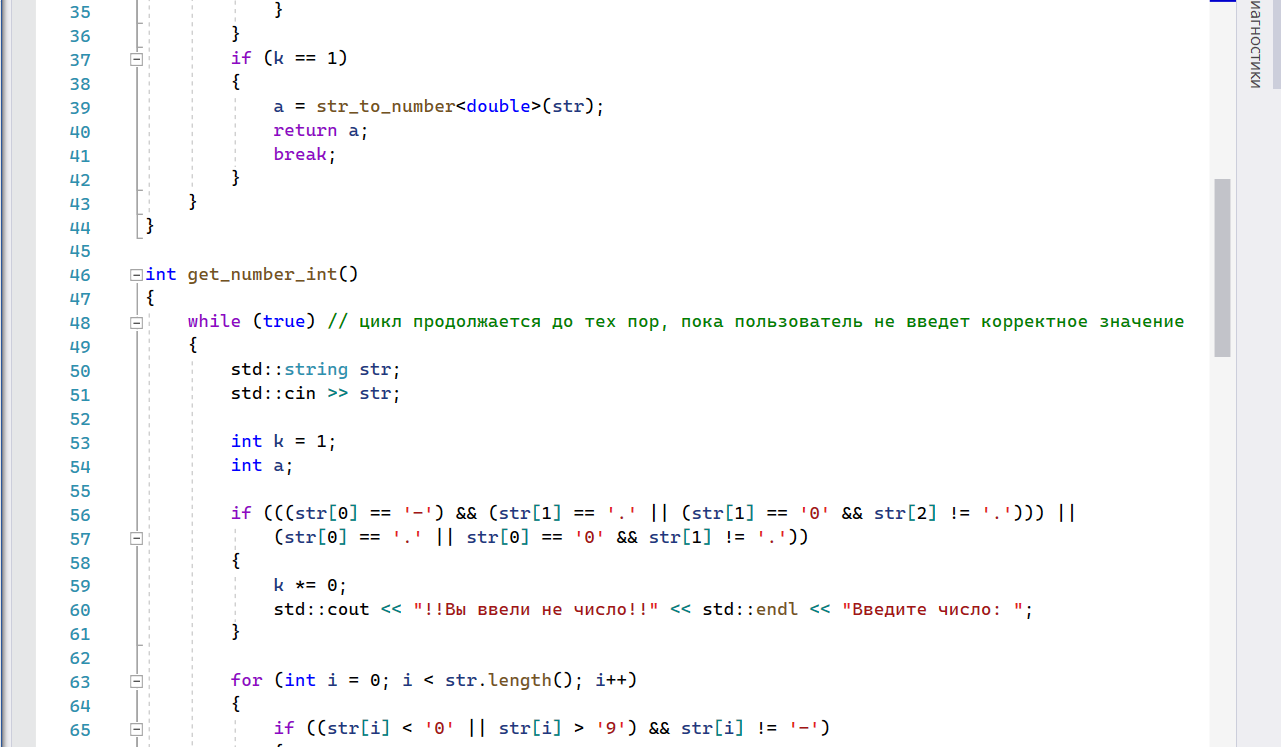


Рисунок 17. Листинг программы various\_functions.h и various\_functions.cpp

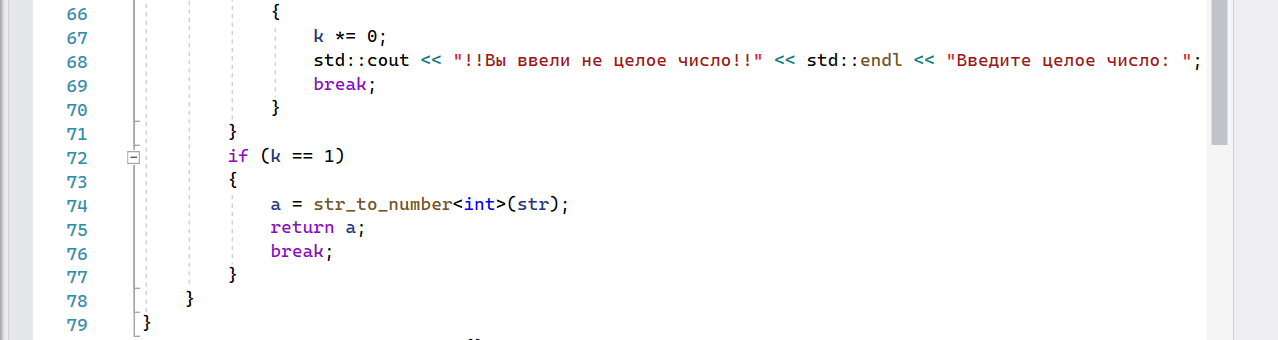


Рисунок 18. Листинг программы various\_functions.h и various\_functions.cpp

# Результат.

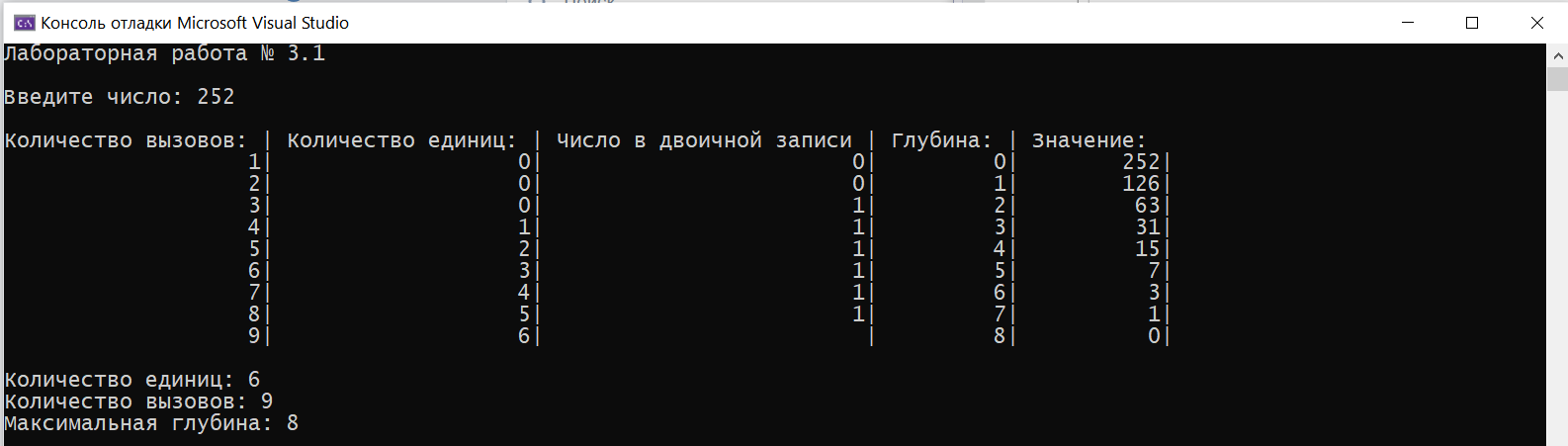


Рисунок 19. Тест

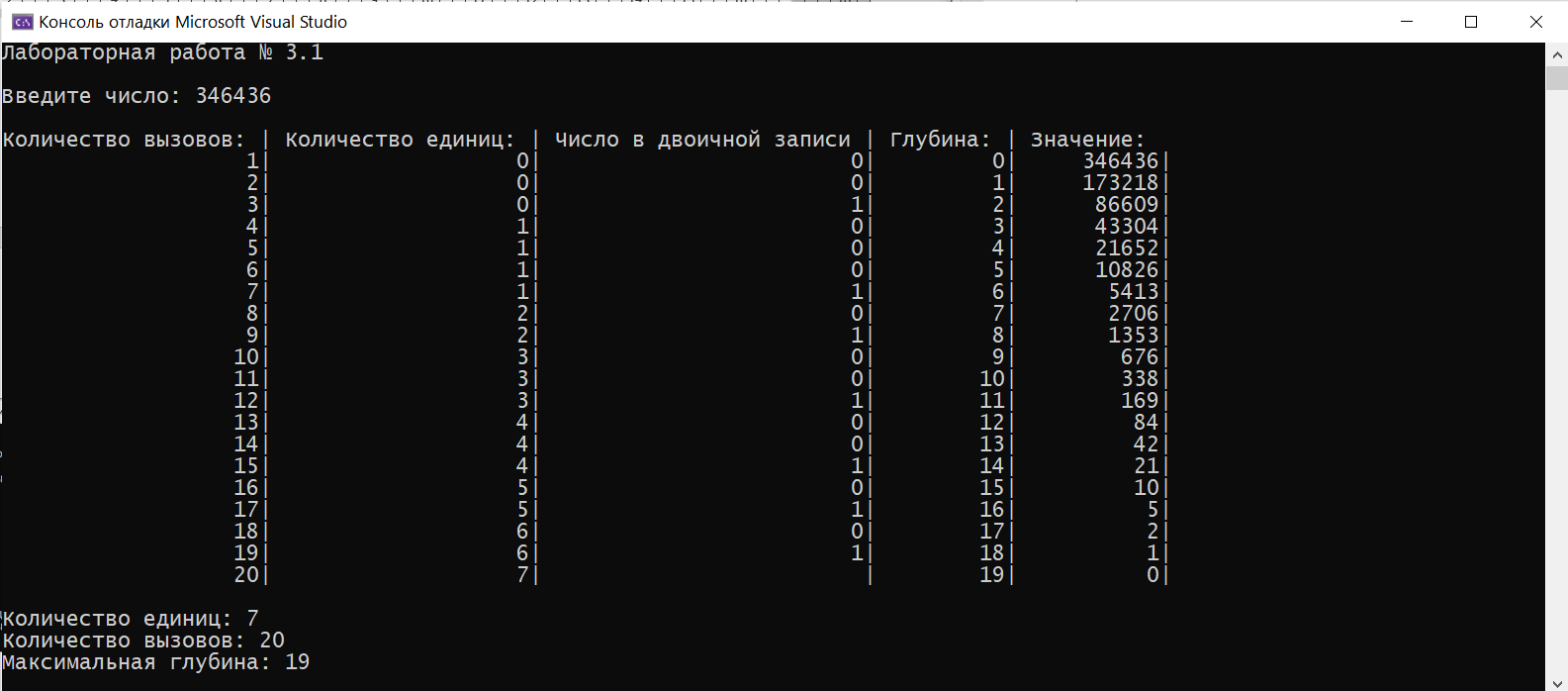


Рисунок 20. Тест

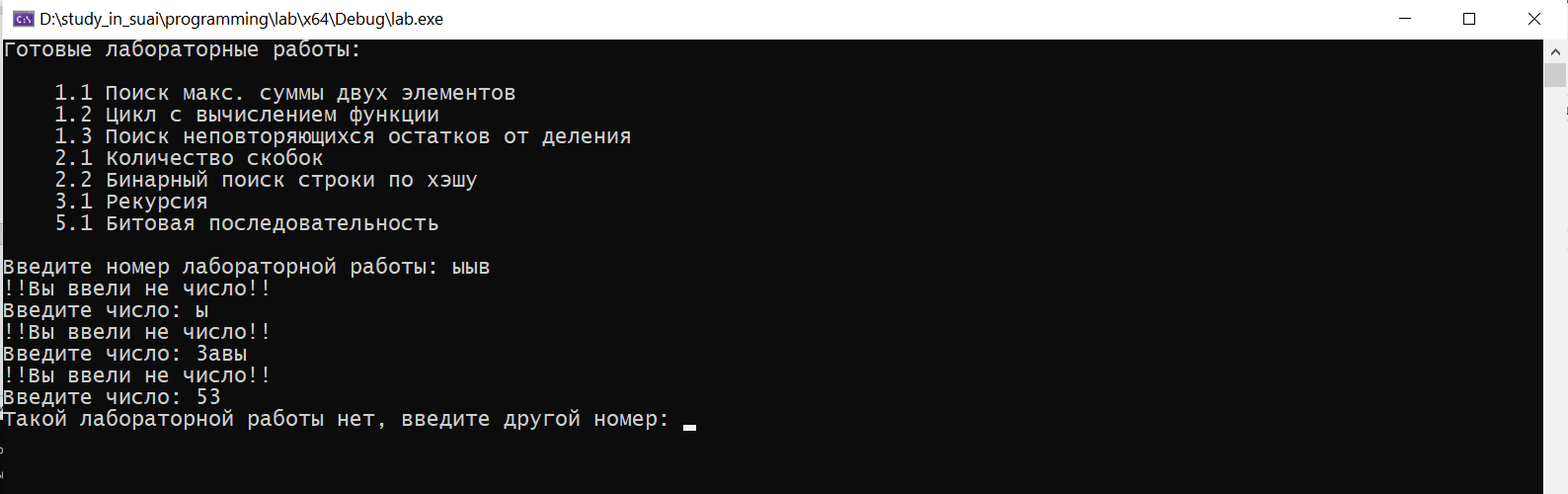


Рисунок 21. Тест

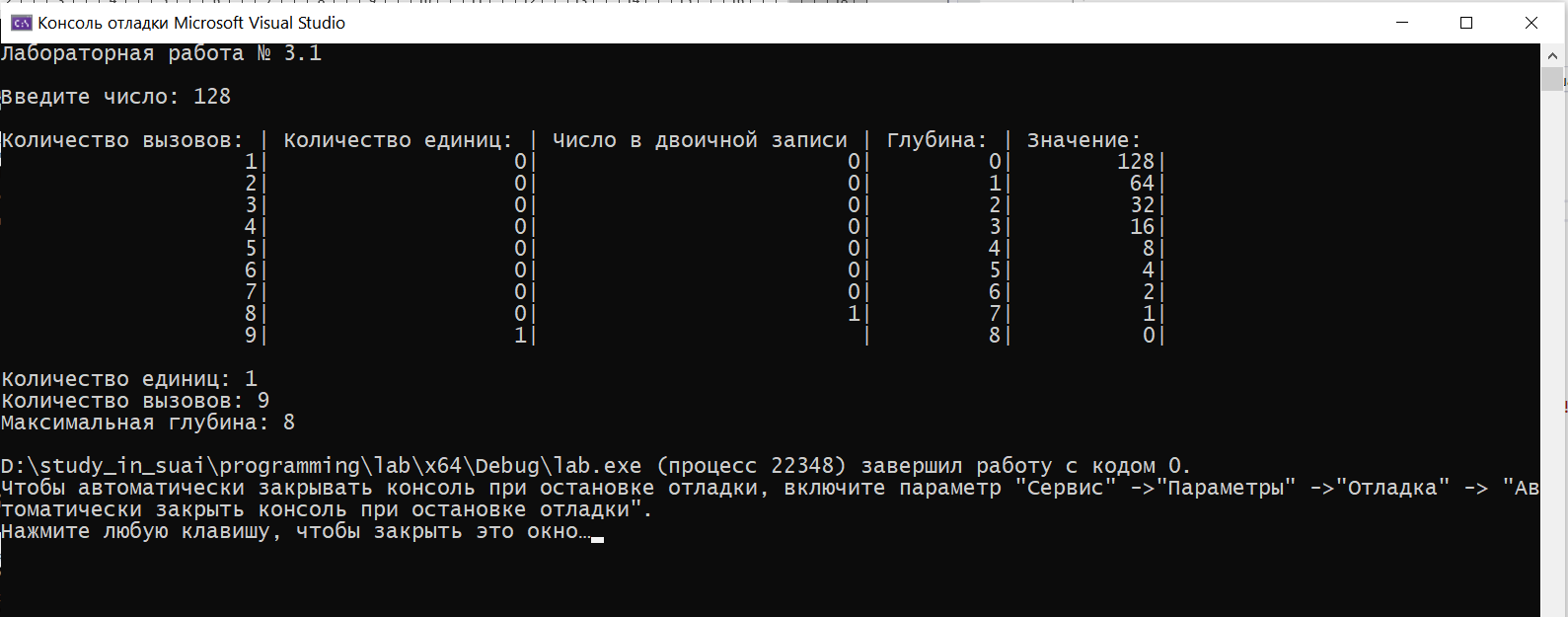


Рисунок 22. Тест

# Вывод.

В ходе выполнения лабораторной работы были освоены:

* массивы
* условные операторы
* циклы
* функции
* вызов функций из другого файла
* обработка ввода символов
* рекурсия
* динамические массивы
* форматированный вывод

Программа принимает число и выводит количество единиц. Программа делит число 2 пока число не станет равно нулю.