|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1** | |
| **по дисциплине** | |
| **«**Структуры и алгоритмы обработки данных**»**  **Тема: «Поразрядные операции и их применение»** | |
|  | |
| Выполнил студент группы ИКБО-04-21 | Исаев В.В. |
| Принял преподаватель | Филатов А.С. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2022

# **Цель работы**

Получение навыков применения поразрядных операций в алгоритмах.

# **Постановка задачи**

1. Разработать программу, которая продемонстрирует выполнение упражнений варианта. Результаты выполнения упражнения выводить на монитор.

Требования к упражнениям:

1. Определить переменную целого типа, присвоить ей значение, используя константу в шестнадцатеричной системе счисления. Разработать функцию, которое установит заданные в задании биты исходного значения переменной в значение 1, используя соответствующую маску и поразрядную операцию.
2. Разработать функцию, которая обнуляет заданные в задании биты исходного значения целочисленной переменной, введенной пользователем, используя соответствующую маску и поразрядную операцию.
3. Разработать функцию, которая умножает значение целочисленной переменной, введенной пользователем, на множитель, используя соответствующую поразрядную операцию.
4. Разработать функцию, которая делит значение целочисленной переменной, введенной пользователем, на делитель, используя соответствующую поразрядную операцию.
5. Разработать функцию, реализующую задание, в которой используются только поразрядные операции. В выражении используется маска – переменная. Маска инициализируется единицей в младшем разряде (маска 1) или единицей в старшем разряде (маска 2). Изменяемое число и n вводится с клавиатуры.
6. Провести тестирование программы на небольших объемах данных, введенных вручную. Разработанные тесты должны покрывать все случаи входных данных (средний, лучший, худший). Результаты тестирования свести в сводные таблицы.
7. Составить отчет, отобразив в нем описание выполнения всех этапов разработки, тестирования и код всей программы со скриншотами результатов тестирования.

Вариант №8. Условие задания:

|  |  |
| --- | --- |
| Упражнение 1 | 1-ый, 6-ой, 9-ый |
| Упражнение 2 | 1-ый, 6-ой, 9-ый |
| Упражнение 3 | 8 |
| Упражнение 4 | 8 |
| Упражнение 5 | Установить n-ый бит в 1, используя маску 1 |

# **Решение**

Поразрядные операции - операции над цепочками битов. Как правило, в этот класс включаются логические побитовые операции и битовые сдвиги. Применяются в языках программирования и цифровой технике, изучаются в дискретной математике.

Битовые операции лежат в основе обработки цифровых сигналов: посредством их из одного или нескольких сигналов на входе получается новый сигнал, который в свою очередь может быть подан на вход одной или нескольким таким операциям. Именно битовые операции в сочетании с запоминающими элементами (например, триггерами) реализуют всё богатство возможностей современной цифровой техники.

Побитовое отрицание (~) – унарная операция. Преобразовывает поданный бит в противоположный, то есть 1 в 0, а 0 в 1.

Таблица 1. Таблица истинности операции Побитовое отрицание

|  |  |  |
| --- | --- | --- |
| **НЕ** | **0** | **1** |
|  | 1 | 0 |

Побитовое И (&) – бинарная операция. В случае если оба бита будут равны 1, то результат будет 1, в противном случае будет 0.

Таблица 2. Таблица истинности операции Побитовое И

|  |  |  |
| --- | --- | --- |
| **И** | **0** | **1** |
| **0** | 0 | 0 |
| **1** | 0 | 1 |

Побитовое ИЛИ (|) – бинарная операция. В случае если хотя бы один бит будет равен 1, то результат будет 1. В противном случае, 0.

Таблица 3. Таблица истинности операции Побитовое ИЛИ

|  |  |  |
| --- | --- | --- |
| **ИЛИ** | **0** | **1** |
| **0** | 0 | 1 |
| **1** | 1 | 1 |

Побитовое исключающее ИЛИ (^) – бинарная операция. В случае, если оба бита одинаковые, результат будет 0, в противном случае будет 1.

Таблица 4. Таблица истинности операции Побитовое исключающее ИЛИ

|  |  |  |
| --- | --- | --- |
| **Искл. ИЛИ** | **0** | **1** |
| **0** | 1 | 0 |
| **1** | 0 | 1 |

Битовый сдвиг (<< - умножение, >> - деление) – бинарная операция. Все биты в двоичной системе счисления сдвигаются влево (умножение на 2) или вправо (деление на 2).

Упражнение 1. В статичном числе 1-ый, 6-ой и 9-ый биты сделать 1.

Для реализации задания была разработана маска в 16-чной системе счисления 0x242. Данная маска и статичное число передаются в разработанный метод func1. Метод возвращает результат операции ,путем сложения числа и маски.

|  |
| --- |
| //функция сложения с помощью поразрядных операций (упр.1)  int func1(int x,int maska)  {  //1,6,9  return x | maska;  } |

Упражнение 2. В статичном числе 1-ый, 6-ой и 9-ый биты сделать 0.

Для реализации задания была разработана маска в 16-чной системе счисления 0xFFFFFDBD. Данная маска и введенное пользователем число передаются в разработанный метод func2. Метод возвращает результат операции, путем умножения числа на маску.

|  |
| --- |
| //функция умножения с помощью поразрядных операций (упр.2)  int func2(int x, int maska)  {  //1,6,9  return x&maska;  } |

Упражнение 3. Умножение числа на 8.

Для реализации задания был разработан метод func3, в который передается введенное пользователем число. Метод возвращает результат операции ,путем побитового сдвига числа.

|  |
| --- |
| //функция поразрядного умножения с помощью побитового умножения(упр.3)  int func3(int x)  {  return x<<3;  } |

Упражнение 4. Деление числа на 8.

Для реализации задания был разработан метод func4, в который передается введенное пользователем число. Метод возвращает результат операции, путем побитового сдвига числа.

|  |
| --- |
| //функция поразрядного деления с помощью побитового деления(упр.4)  int func4(int x)  {  return x>>3;  } |

Упражнение 5. Установить введенный пользователем бит в 1, используя маску 0x1.

Для реализации задания был разработан метод func5, в который передается введенное пользователем число, маска и номер бита. Метод возвращает результат операции, путем сдвига маски и последующего сложения числа и новой маски.

|  |
| --- |
| //функция двух последующих бинарных операций (упр. 5)  int func5(int x, int maska, int n)  {  maska = maska << n;  return x | maska;  } |

После запуска программы, пользователю предлагается выбрать номер упражнения в меню. После выбора, пользователю выводится информация о том, какие переменные ему нужно ввести. В случае правильного ввода всех данных ,программа выдает результат упражнения. Для повторного выполнения упражнения, программу необходимо закрыть и запустить заново.

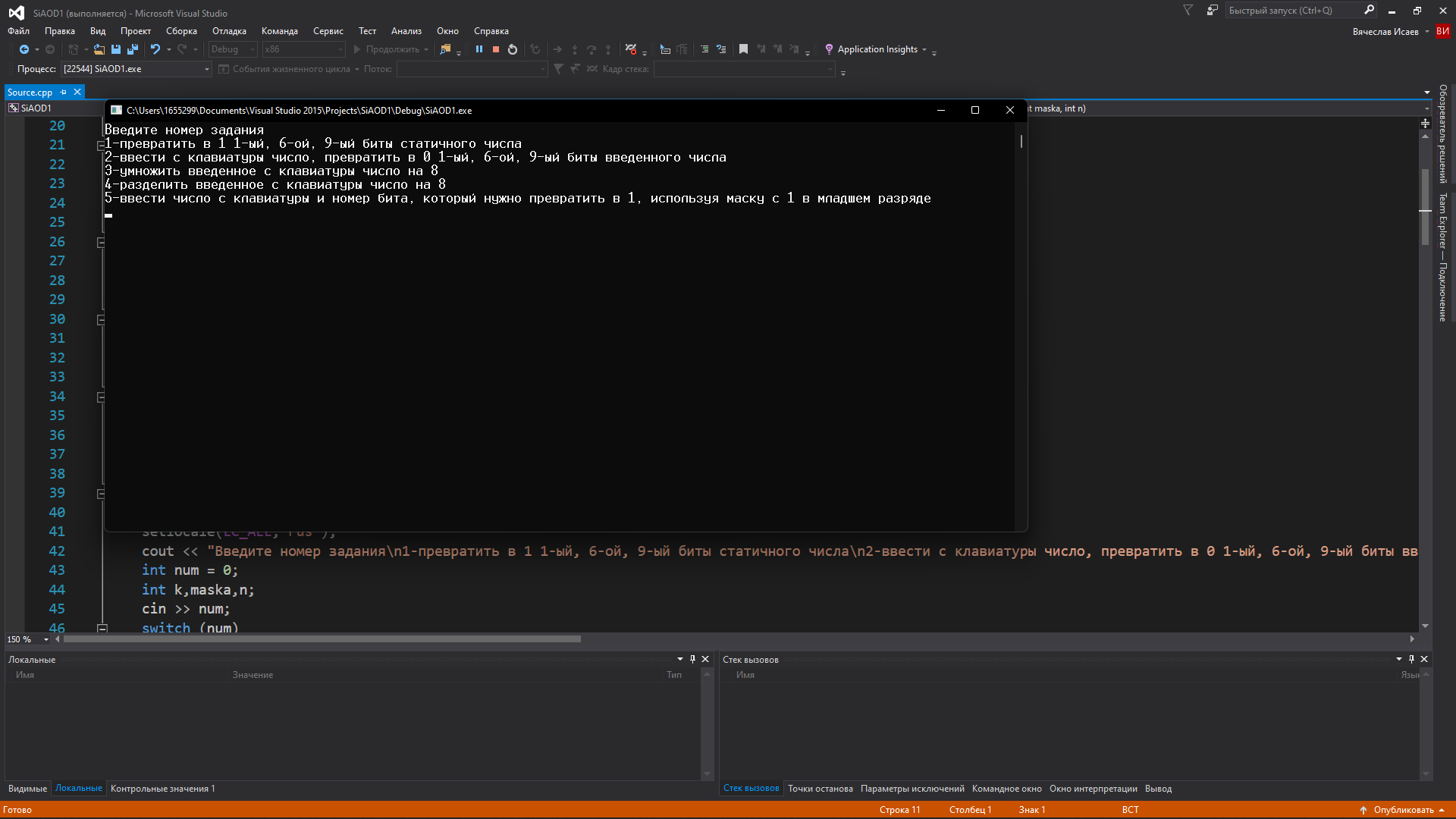


Рисунок 1. Меню программы

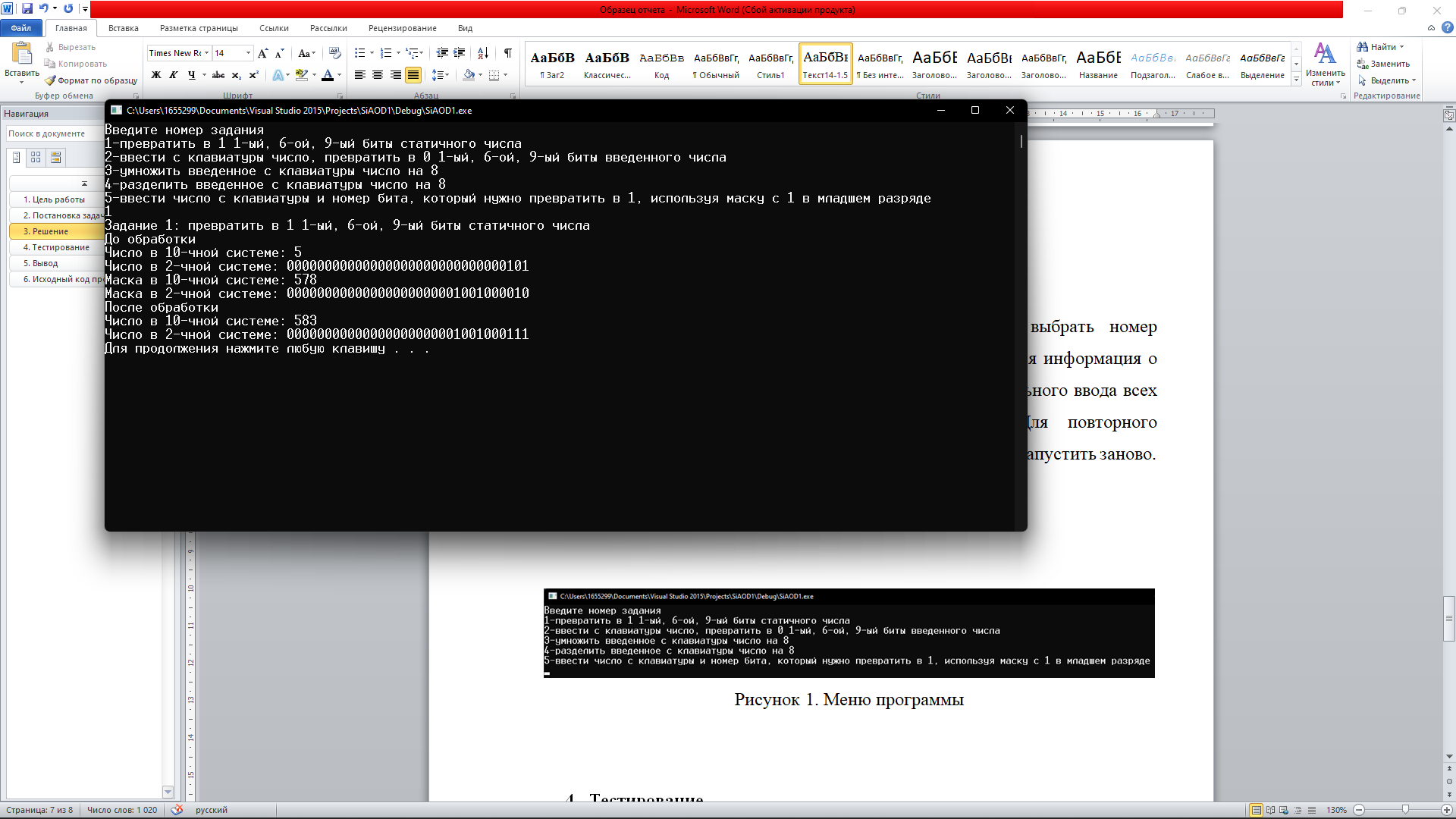


Рисунок 2. Вид задания без пользовательского ввода

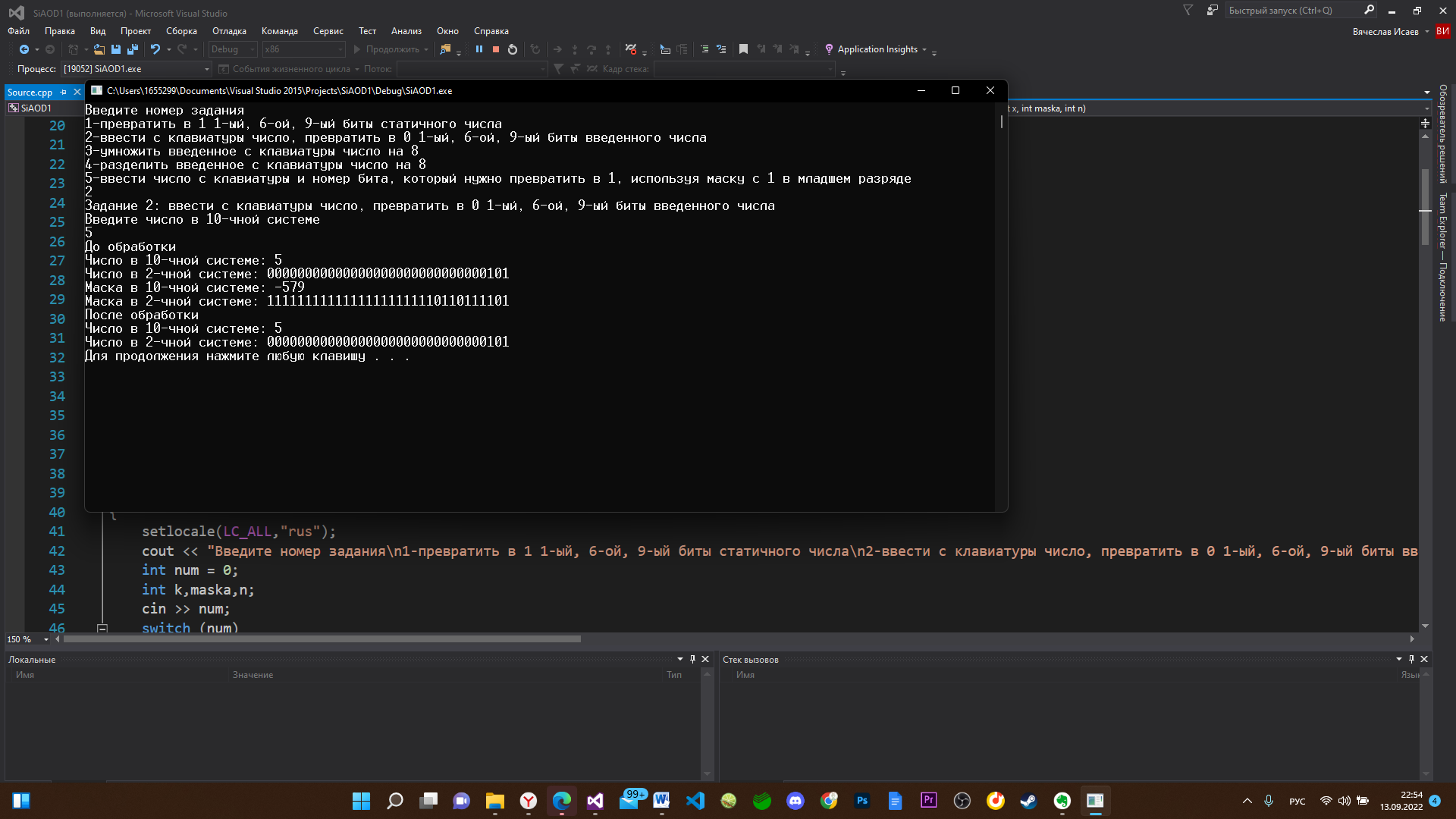


Рисунок 3. Вид задания с пользовательским вводом

# **Тестирование**

Для тестирования программы были выбраны следующие числа в 10-чной системе счисления.

Таблица 5. Числа, выбранные для тестирования программы

|  |  |  |
| --- | --- | --- |
| **Порядковый номер тестирования** | **Номер упражнения** | **Число** |
| 1 | 1 | Не вводится |
| 2 | 2 | 9999999999999999999 |
| 3 | 2 | 5 |
| 4 | 3 | 6 |
| 5 | 3 | 0 |
| 6 | 4 | 24 |
| 7 | 4 | 7 |
| 8 | 5 | 99999;5 |
| 9 | 5 | 15;1 |

Тестирование программы покрывает все входные данные, так как числа выбранные для тестирования, позволяют убедиться, что число не теряет свой вид после применения маски на те биты, которые не должны меняться.

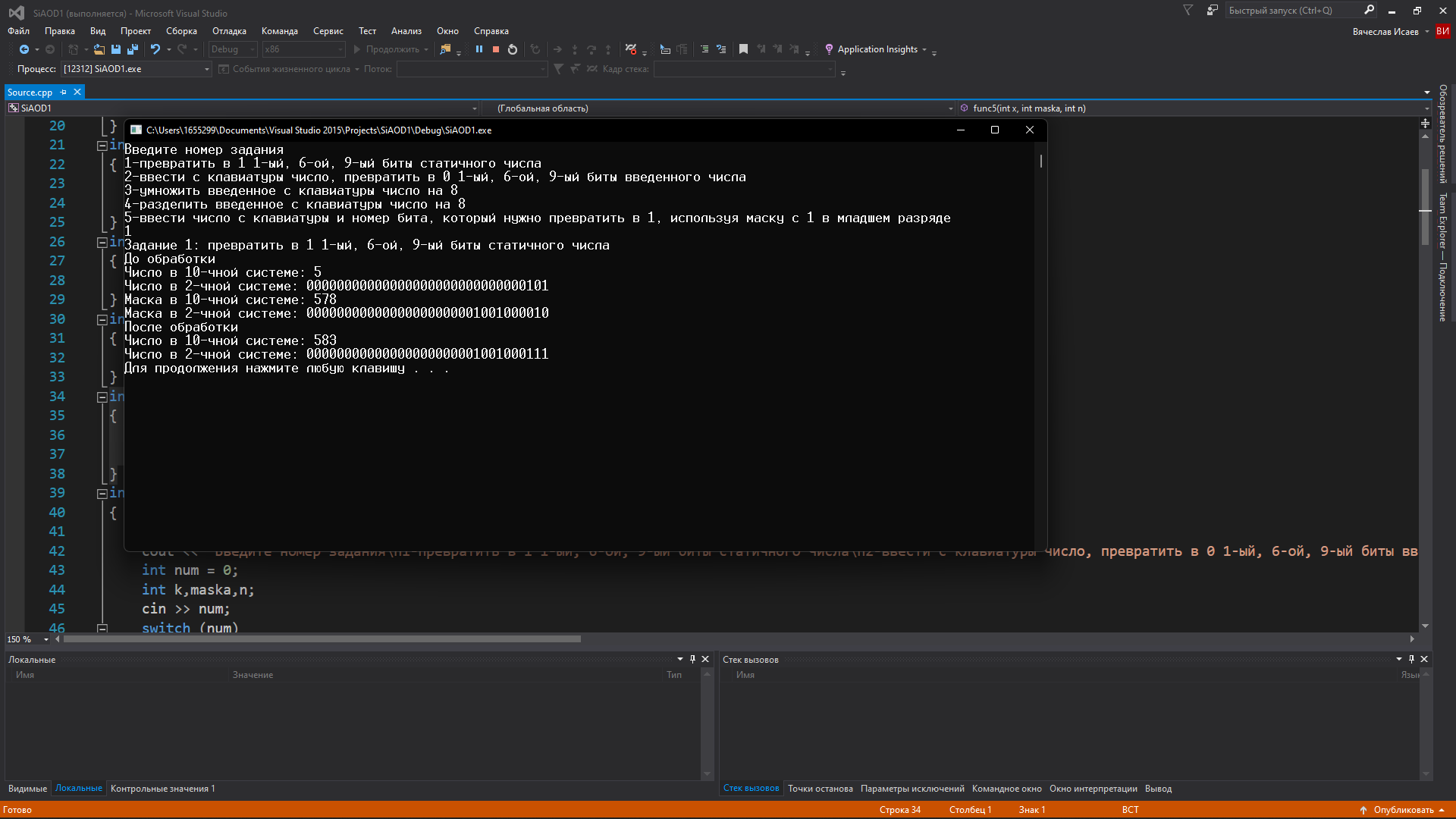


Рисунок 4. 1-е тестирование

Как видно из изображения 4, 1-ый, 6-ой и 9-ый биты входного числа превратились в 1, что увеличило исходное число на 578 в 10-чной системе счисления.

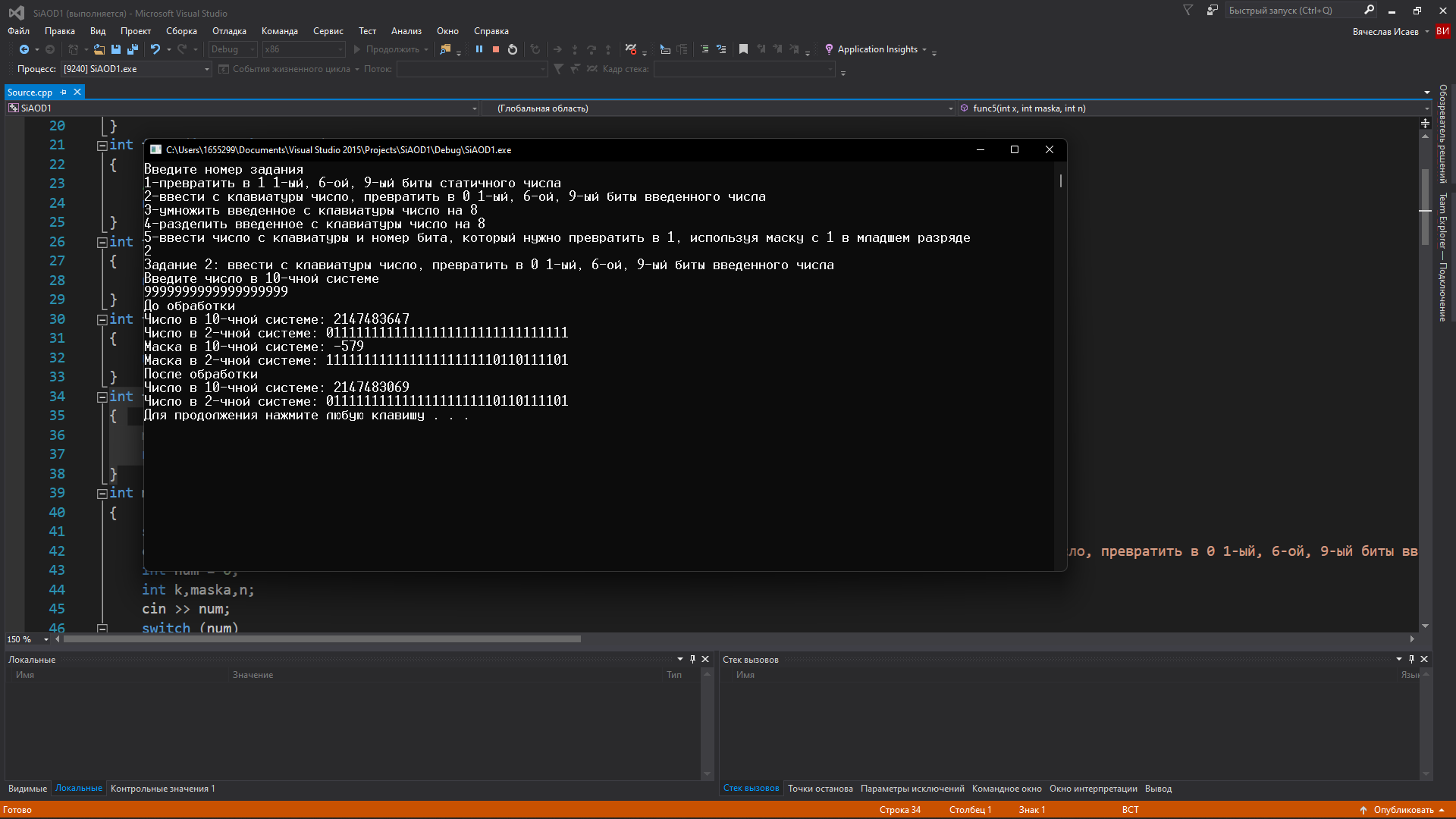


Рисунок 5. 2-е тестирование

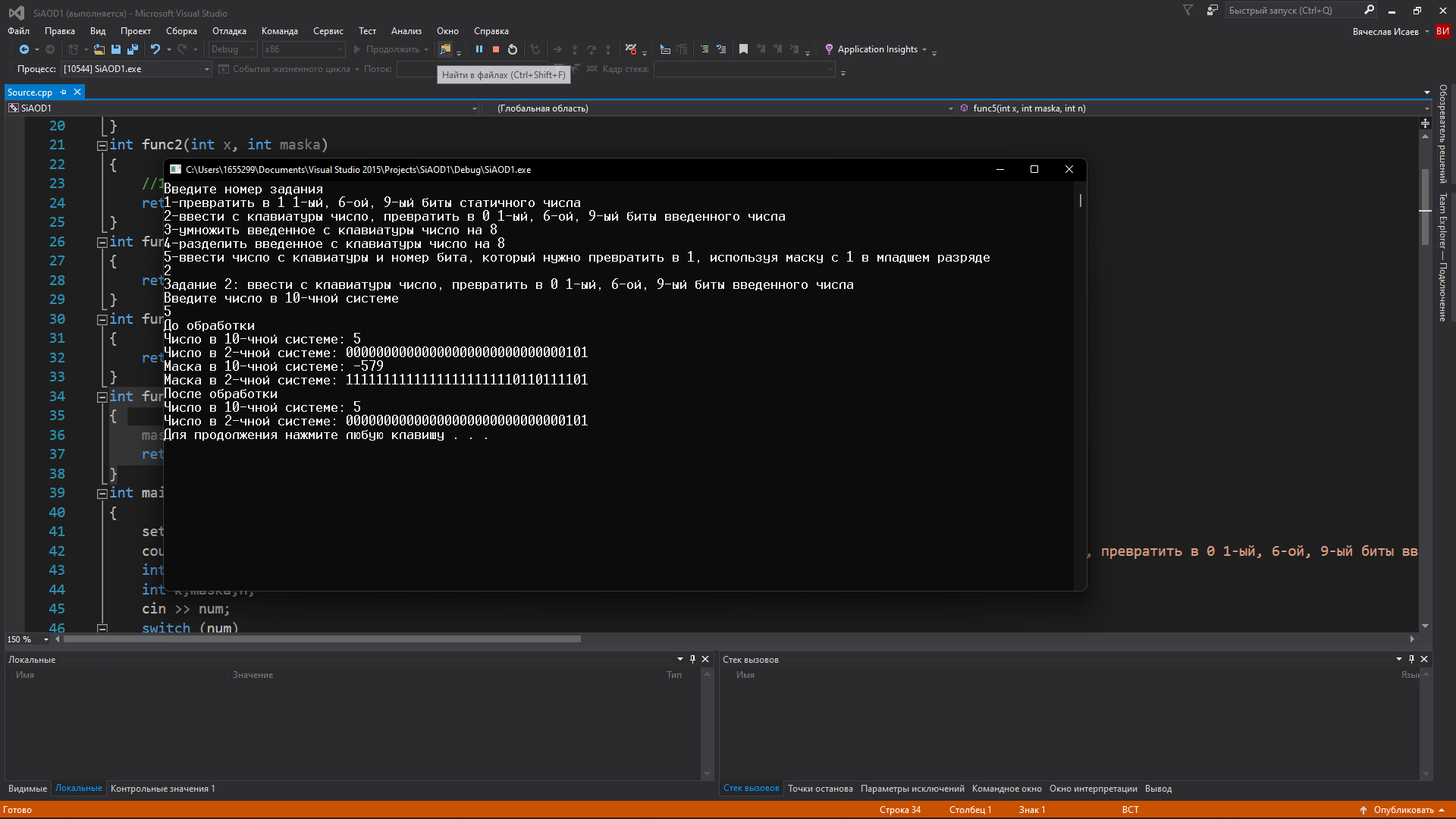


Рисунок 6. 3-е тестирование

Как видно из изображений 5 и 6, в случае, если в исходном числе 1-ый, 6-ой и 9-ый биты и были 0, число не меняется, а в случае, если биты в исходном числе были 1, число уменьшается на 578.

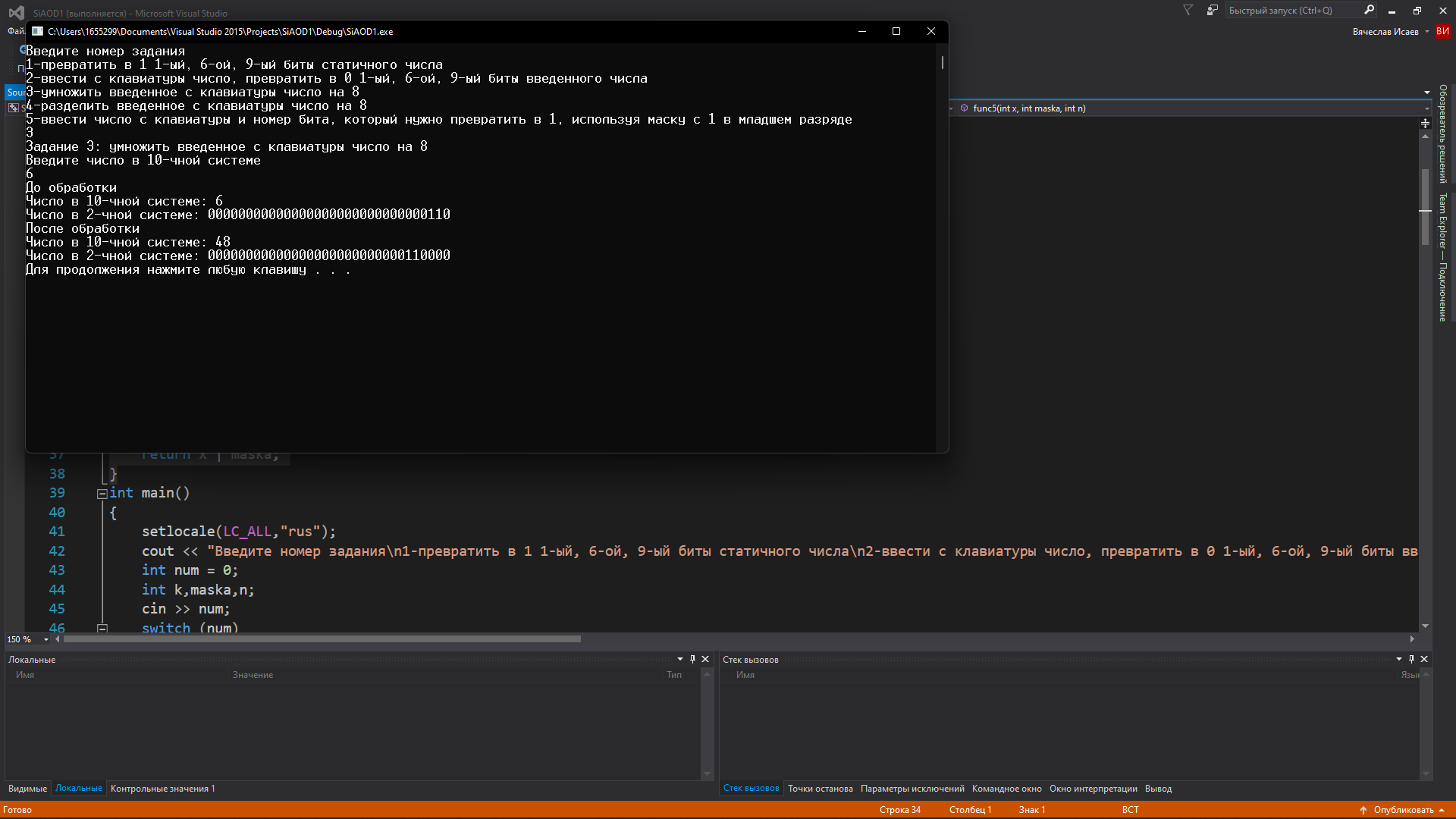


Рисунок 7. 4-е тестирование

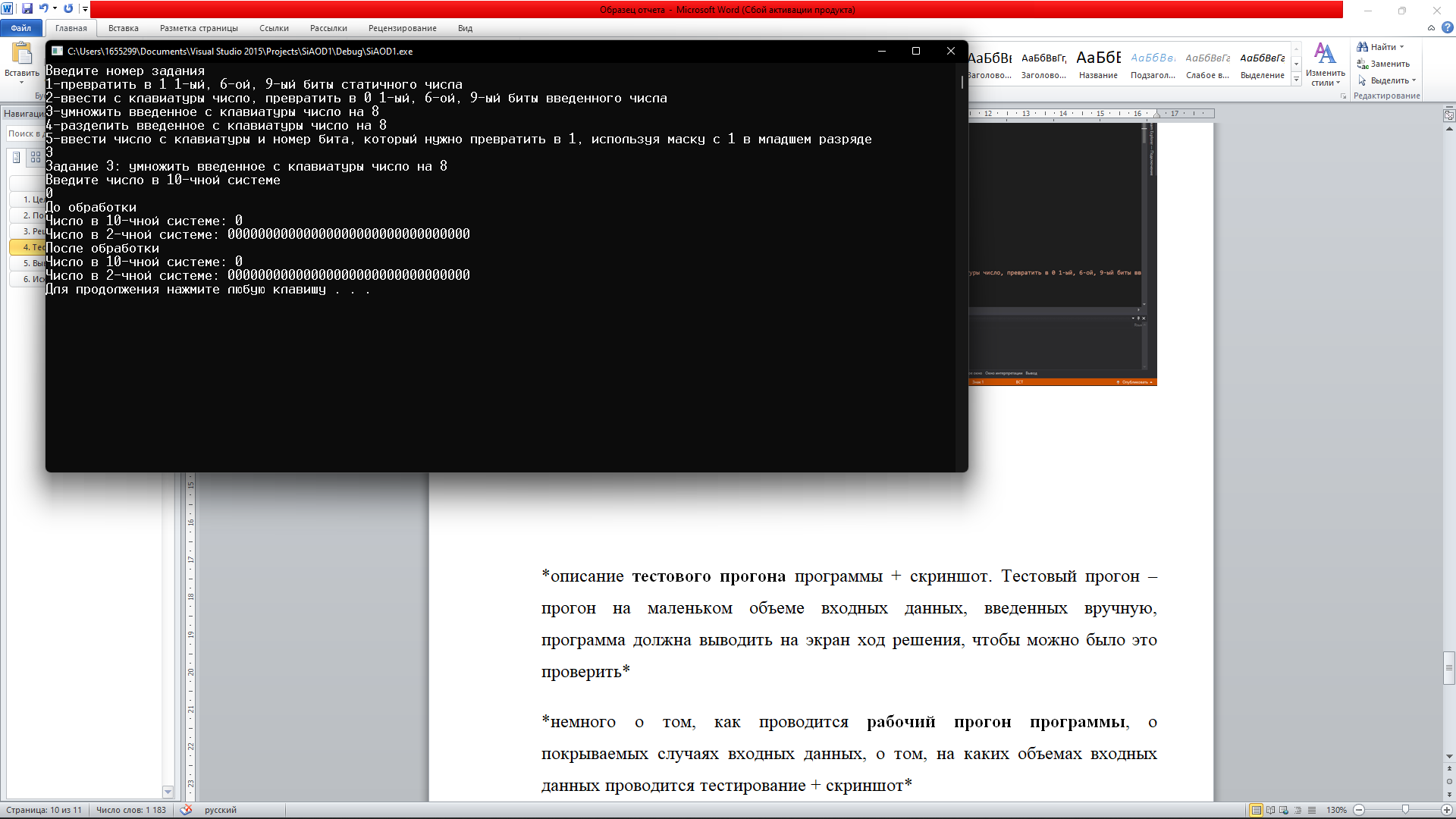


Рисунок 8. 5-е тестирование

Как видно из изображений 7 и 8, исходное число умножается на 8.

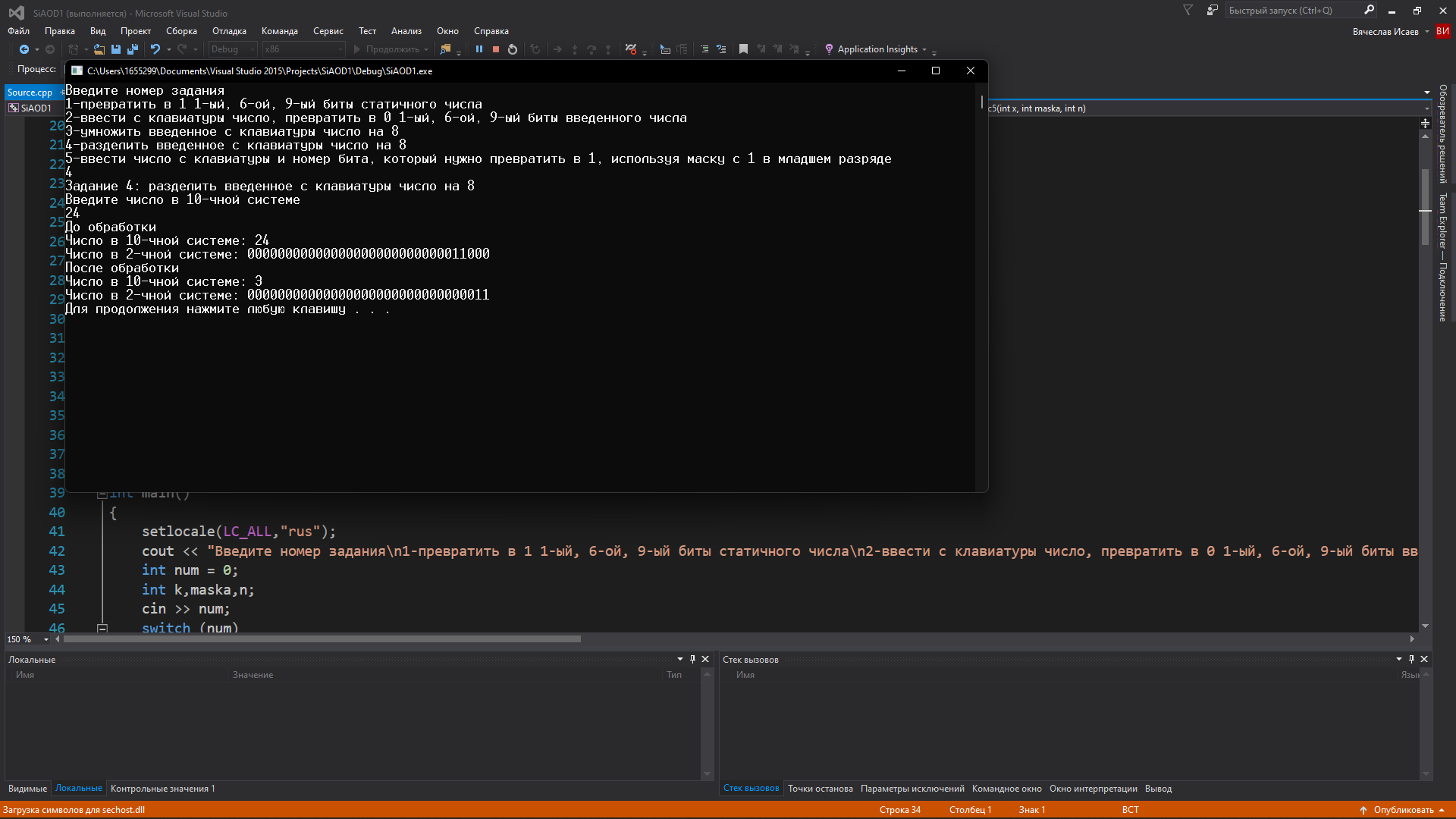


Рисунок 9. 6-е тестирование

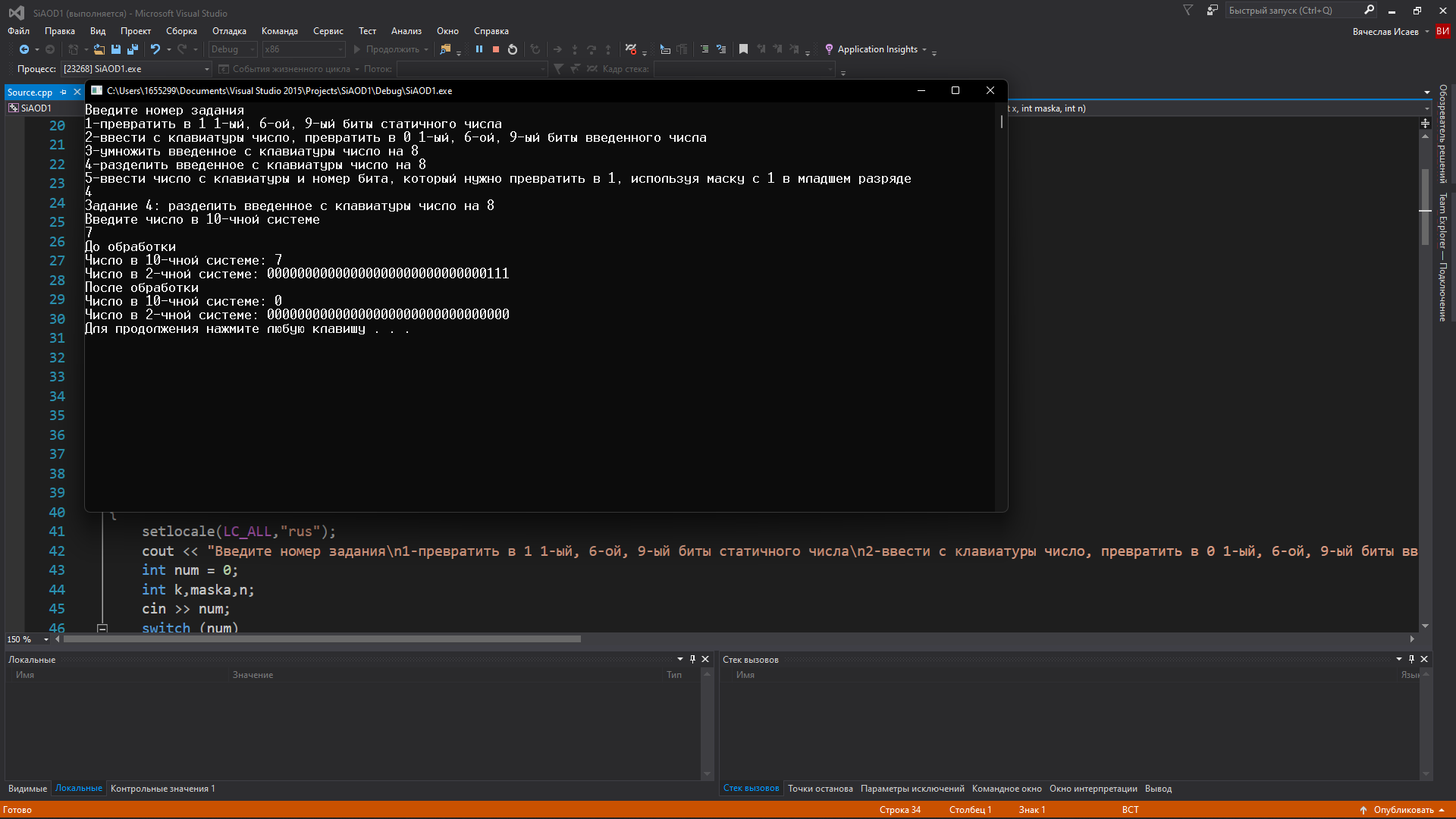


Рисунок 10. 7-е тестирование

Как видно из изображений 9 и 10 число делится на 8 без остатка с округлением в меньшую сторону. В случае, если число меньше 8, результат становится 0.

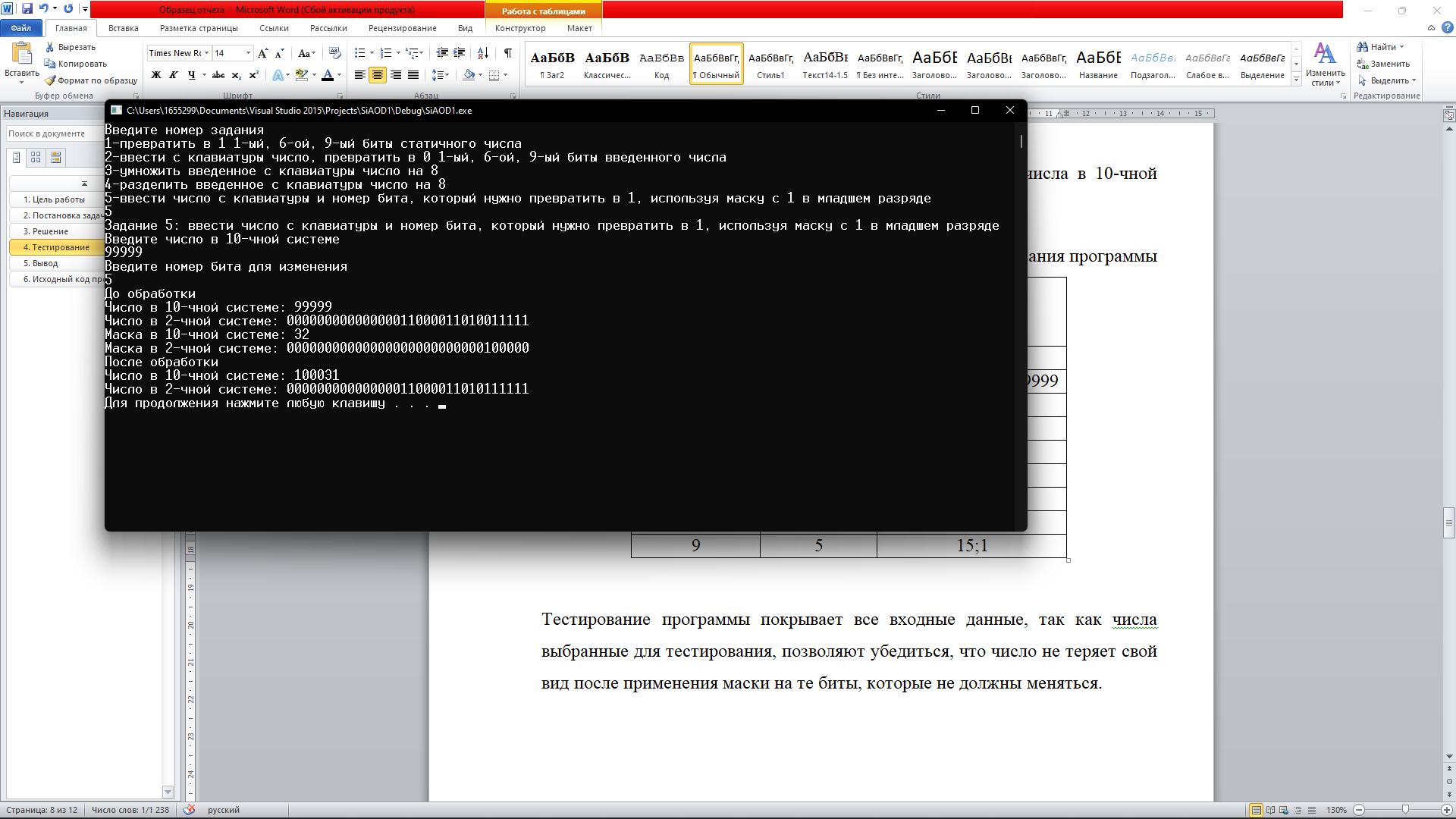


Рисунок 11. 8-е тестирование

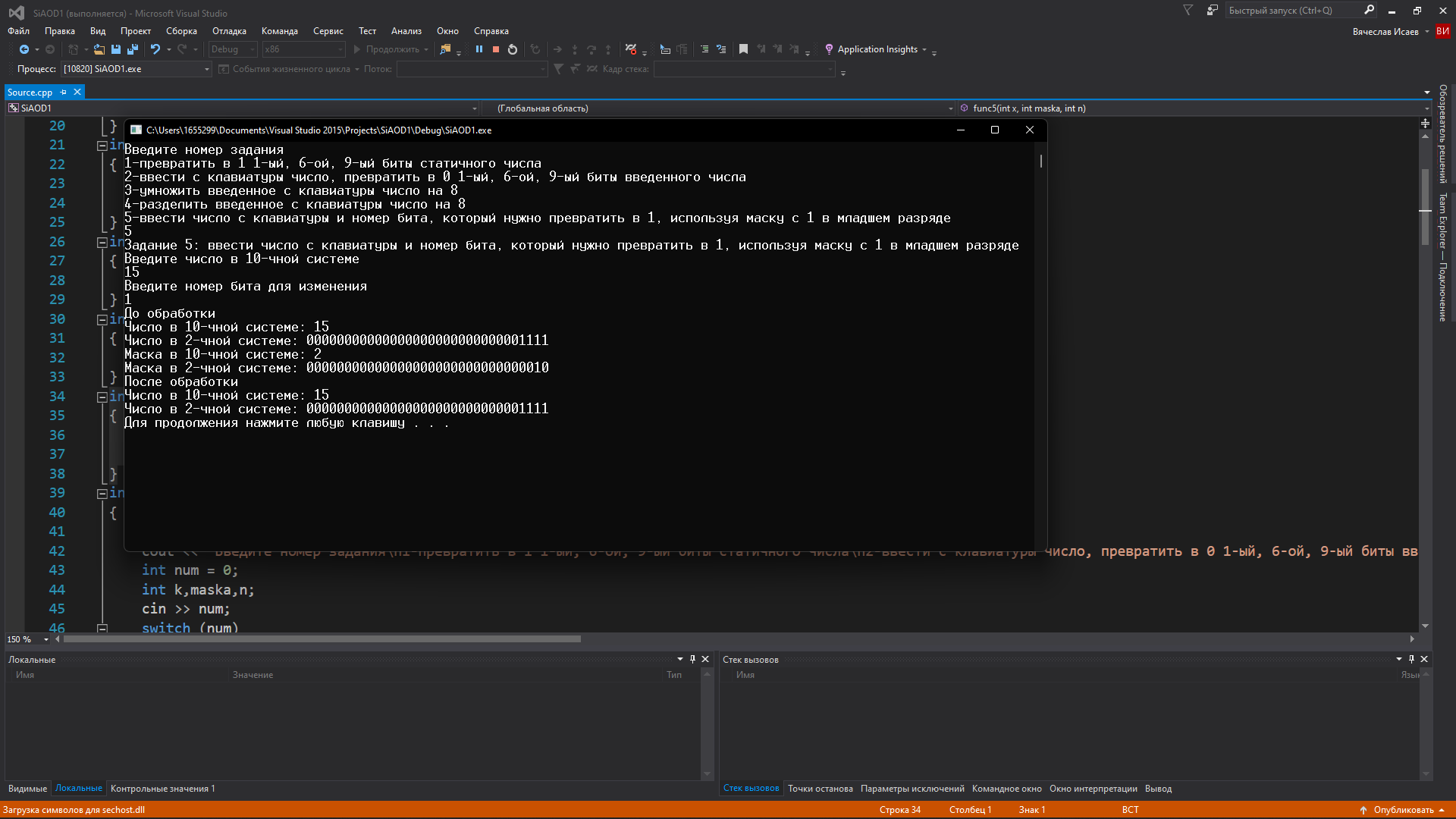


Рисунок 12. 9-е тестирование

Как видно из изображений 11 и 12, если изначальный бит числа был 0, то число увеличивается на 2 в такой степени, номер бита которой был введен. Если бит изначально был 1, то число не меняется.

Из результатов выполнения программы видно:

1. Программа работает корректно.
2. Программа обрабатывает все введенные данные

# **Вывод**

В результате выполнения работы я:

1. Освоил алгоритмы работы с поразрядными операциями и их реализацию на языке программирования C++.
2. Провел тестирование входных данных.

# **Исходный код программы**

|  |
| --- |
| #include <iostream>  using namespace std;  void coutp(unsigned int x)  {  int n = sizeof(int) \* 8;  unsigned maska = (1 << (n - 1));  for (int i = 1; i <= n; i++)  {  cout << ((x&maska) >> (n - i));  maska = maska >> 1;  }  }  int func1(int x,int maska)  {  //1,6,9  return x | maska;  }  int func2(int x, int maska)  {  //1,6,9  return x&maska;  }  int func3(int x)  {  return x<<3;  }  int func4(int x)  {  return x>>3;  }  int func5(int x, int maska, int n)  {  maska = maska << n;  return x | maska;  }  int main()  {  setlocale(LC\_ALL,"rus");  cout << "Введите номер задания\n1-превратить в 1 1-ый, 6-ой, 9-ый биты статичного числа\n2-ввести с клавиатуры число, превратить в 0 1-ый, 6-ой, 9-ый биты введенного числа\n3-умножить введенное с клавиатуры число на 8\n4-разделить введенное с клавиатуры число на 8\n5-ввести число с клавиатуры и номер бита, который нужно превратить в 1, используя маску с 1 в младшем разряде" << endl;  int num = 0;  int k,maska,n;  cin >> num;  switch (num)  {  case 1:  cout << "Задание 1: превратить в 1 1-ый, 6-ой, 9-ый биты статичного числа" << endl;  k = 5;  maska = 0x242;  cout << "До обработки" << endl;  cout << "Число в 10-чной системе: " << k << endl;  cout << "Число в 2-чной системе: ";  coutp(k);  cout << endl;  cout << "Маска в 10-чной системе: " << maska << endl;  cout << "Маска в 2-чной системе: ";  coutp(maska);  cout << endl;  cout << "После обработки" << endl;  cout << "Число в 10-чной системе: " << func1(k, maska) << endl;  cout << "Число в 2-чной системе: ";  coutp(func1(k,maska));  cout << endl;  break;  case 2:  cout << "Задание 2: ввести с клавиатуры число, превратить в 0 1-ый, 6-ой, 9-ый биты введенного числа" << endl;  cout << "Введите число в 10-чной системе" << endl;  cin>>k;  maska = 0xFFFFFDBD;  cout << "До обработки" << endl;  cout << "Число в 10-чной системе: " << k << endl;  cout << "Число в 2-чной системе: ";  coutp(k);  cout << endl;  cout << "Маска в 10-чной системе: " << maska << endl;  cout << "Маска в 2-чной системе: ";  coutp(maska);  cout << endl;  cout << "После обработки" << endl;  cout << "Число в 10-чной системе: " << func2(k, maska) << endl;  cout << "Число в 2-чной системе: ";  coutp(func2(k, maska));  cout << endl;  break;  case 3:  cout << "Задание 3: умножить введенное с клавиатуры число на 8" << endl;  cout << "Введите число в 10-чной системе" << endl;  cin >> k;  cout << "До обработки" << endl;  cout << "Число в 10-чной системе: " << k << endl;  cout << "Число в 2-чной системе: ";  coutp(k);  cout << endl;  cout << "После обработки" << endl;  cout << "Число в 10-чной системе: " << func3(k) << endl;  cout << "Число в 2-чной системе: ";  coutp(func3(k));  cout << endl;  break;  case 4:  cout << "Задание 4: разделить введенное с клавиатуры число на 8" << endl;  cout << "Введите число в 10-чной системе" << endl;  cin >> k;  cout << "До обработки" << endl;  cout << "Число в 10-чной системе: " << k << endl;  cout << "Число в 2-чной системе: ";  coutp(k);  cout << endl;  cout << "После обработки" << endl;  cout << "Число в 10-чной системе: " << func4(k) << endl;  cout << "Число в 2-чной системе: ";  coutp(func4(k));  cout << endl;  break;  break;  case 5:  cout << "Задание 5: ввести число с клавиатуры и номер бита, который нужно превратить в 1, используя маску с 1 в младшем разряде" << endl;  cout << "Введите число в 10-чной системе" << endl;  cin >> k;  cout << "Введите номер бита для изменения" << endl;  cin >> n;  maska = 0x1;  cout << "До обработки" << endl;  cout << "Число в 10-чной системе: " << k << endl;  cout << "Число в 2-чной системе: ";  coutp(k);  cout << endl;  cout << "Маска в 10-чной системе: " << (maska << n) << endl;  cout << "Маска в 2-чной системе: ";  coutp(maska << n);  cout << endl;  cout << "После обработки" << endl;  cout << "Число в 10-чной системе: " << func5(k, maska, n) << endl;  cout << "Число в 2-чной системе: ";  coutp(func5(k, maska, n));  cout << endl;  break;  default:  cout << "Неправильный номер"<<endl;  system("pause");  exit(0);  break;  }  system("pause");  } |