**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №1**

**по дисциплине «Программирование»**

Тема: Типы данных, определяемые пользователем. Структуры

|  |  |  |
| --- | --- | --- |
| Студент гр. 5372 |  | Павленко П. С. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2025

**Цель работы.**

Изучить внутреннее представление данных различных типов в памяти компьютера, освоить методы анализа структуры числовых и логических типов, а также научиться работать с побитовыми операциями для визуализации и изменения представления данных.

**Основные теоретические положения.**

1. Типы данных в языке программирования

Каждый тип данных (целые, вещественные, символьные, логические) имеет фиксированный размер в памяти, который зависит от архитектуры компьютера и компилятора.

Размер типов определяется в байтах, что позволяет судить о диапазоне значений.

2. Внутреннее представление целых чисел

Целые числа хранятся в двоичной системе.

Используется дополнительный код для хранения отрицательных значений.

Существуют знаковый бит и значащие разряды.

3. Представление вещественных чисел (float, double, long double)

Хранятся по стандарту IEEE 754: число разбивается на знак, порядок и мантиссу.

Точность и диапазон зависят от длины мантиссы и экспоненты.

4. Логический и символьный типы

bool хранит значение «истина/ложь» в минимальном числе битов.

char хранит код символа (обычно в таблице ASCII или Unicode).

5. Побитовые операции

Позволяют изменять отдельные биты числа.

Применяются для низкоуровневого программирования, оптимизации и анализа представления данных.

**Постановка задачи.**

1. Определить количество памяти (в байтах), выделяемое под разные типы данных: int, short int, long int, float, double, long double, char, bool.

2. Вывести на экран двоичное представление целого числа с выделением знакового и значащих разрядов.

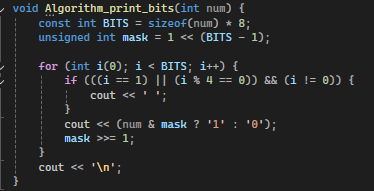
3. Вывести на экран двоичное представление числа типа float, обозначив знак, мантиссу и порядок.

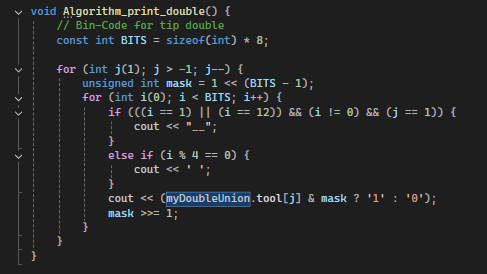
4. Вывести на экран двоичное представление числа типа double, обозначив знак, мантиссу и порядок.

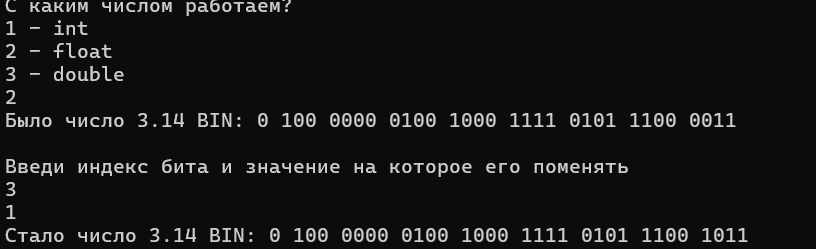
5. Реализовать возможность изменения любого бита введённого числа с помощью логических операций.

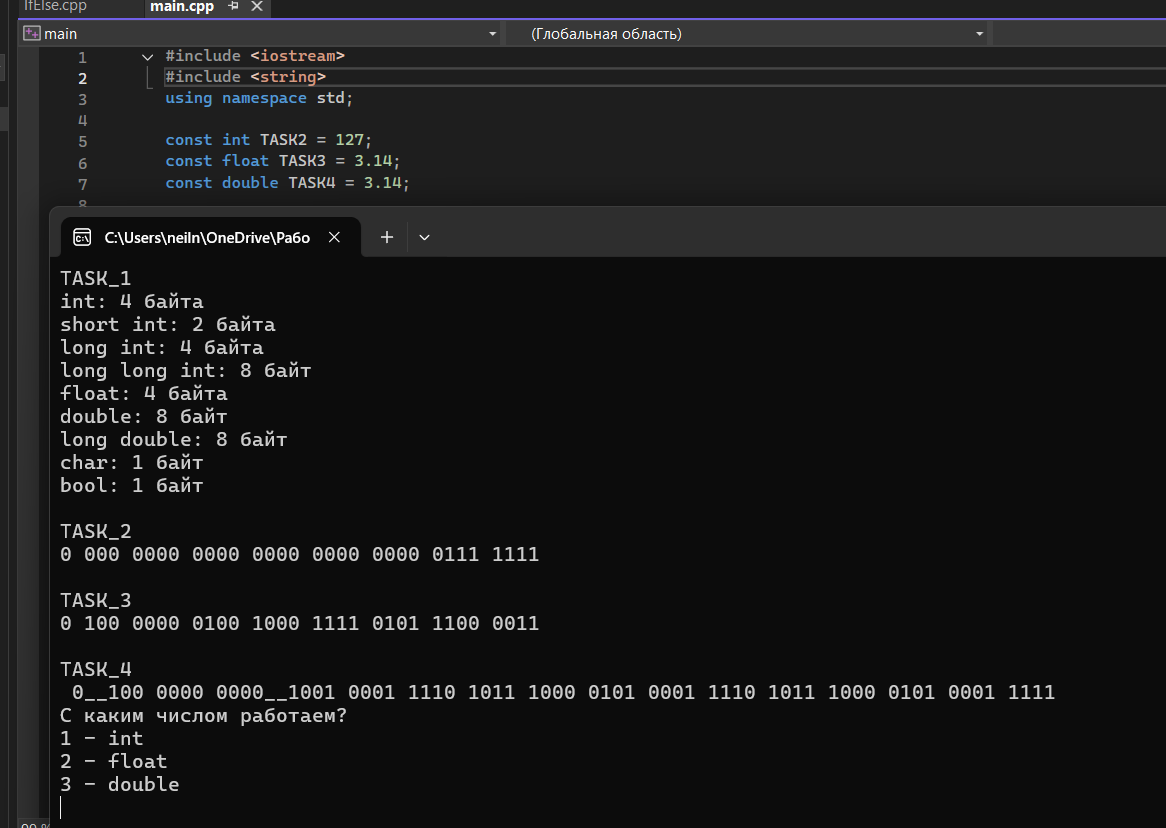
**Выполнение работы.**

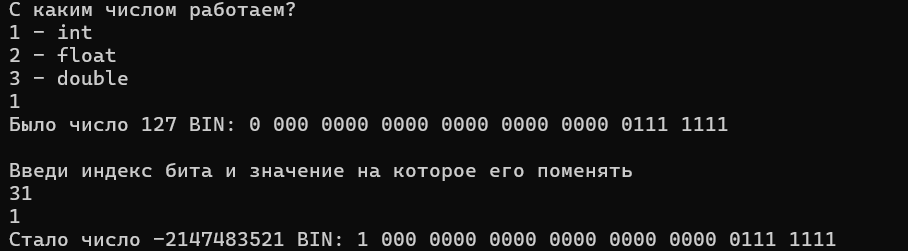
Код программы представлен в приложении А.

В коде использовал 3 основных алгоритма:

1. Алгоритм, выводящий на экран двоичное представление в памяти целого числа
2. Алгоритм, выводящий на экран двоичное представление в памяти типа данных double.
3. Алгоритм, для выполнения 5 задания – «Реализовать возможность произвольного изменения любого бита в введенном числе (для всех типов данных) с использованием логических операций.» На деле – это часть полного алгоритма выполнения 5 задания, которая вынесена в отдельную функцию для лучшей читаемости кода. Algorithm\_for\_task\_5

Блок скриншотов работы программы





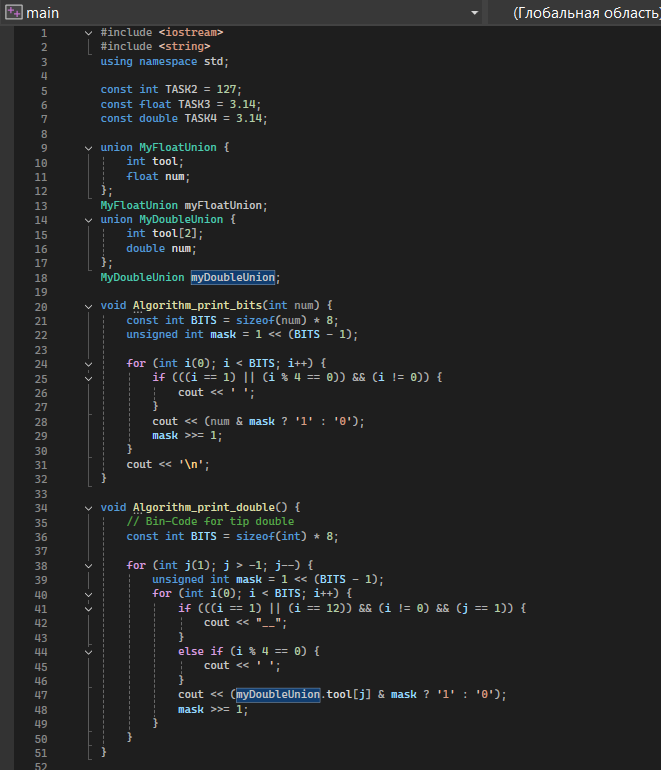
**Выводы.**

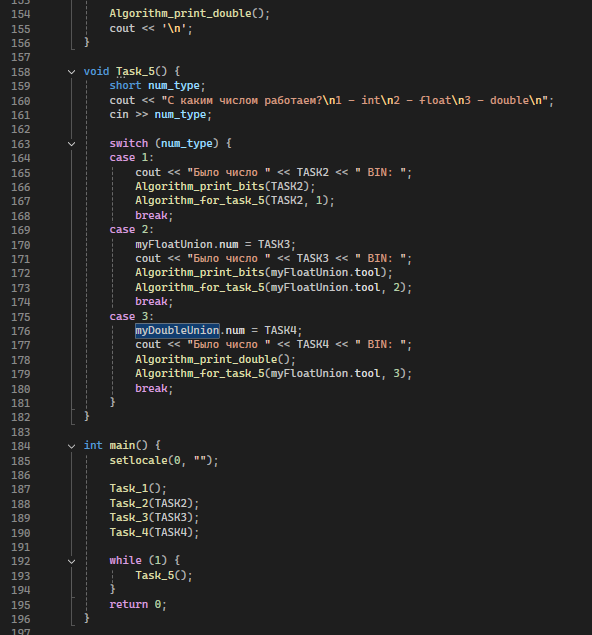
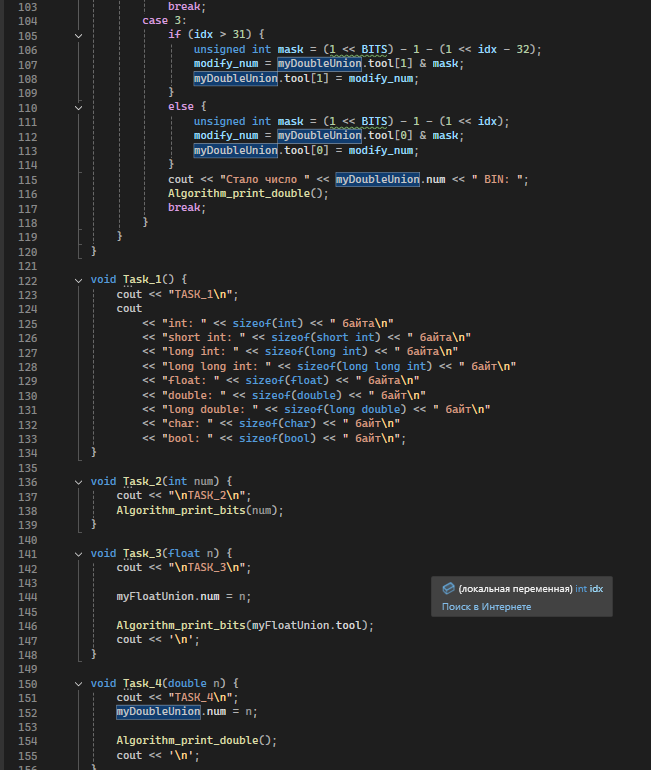
В ходе выполнения практической работы было изучено внутреннее представление различных типов данных в памяти компьютера. Определены размеры стандартных типов (int, short, long, float, double, long double, char, bool) в байтах. На практике реализован вывод двоичного представления целых и вещественных чисел, что позволило наглядно увидеть структуру хранения: знаковый бит, порядок и мантиссу.

Также были применены побитовые операции для изменения отдельных разрядов чисел, что дало возможность лучше понять принципы работы с низкоуровневым представлением данных.

Работа показала важность знания внутреннего устройства типов данных для оптимизации программ, правильного выбора типа при разработке и понимания ограничений вычислительной техники.

Приложение А

рабочий код 



«Выполеннеие работы»

|  |  |
| --- | --- |
| Запуск кода | При запуске кода выводятся ответы на первые 4 задачи. В вечном цикле крутится функция 5-ой задачи. |
| Константы – переменные в коде |  |
| Union MyDoubleUnion; | Объединение для работы с типом double |
| Union MyFloatUnion; | Объединение для работы с типом float |
| Функция TASK\_1 | Функция просто выводит размер типов данных |
| Функция TASK\_2 | Функция-декор выводит двоичное представление в памяти целого числа. Алгоритм вывода записан в функцию Algorithm\_print\_bits |
| Функция Algorithm\_print\_bits | Принимает на вход целое число и при помощи побитового сдвига маски разряд за разрядом выводит биты числа в двоичном представлении. Так же ставит пробелы в нужных местах. |
| Функция TASK\_3 | Функция работает на основе union (объединений), благодаря которым можно получить двоичное представление вещественного типа данных float черезх целое число int и вывести по алгоритму для целых чисел. |
| Функция TASK\_4 | Функция – декор. Основной алгоритм в функции Algorithm\_print\_double |
| Функция Algorithm\_print\_double | Функция выводит биты из 2х чисел массива в union (объединения). Выводит двоичное представление типа double |
| Функция Task\_5 | Функция – менеджер для работы с любым битом любого числа. Часть функционала в функции Algorithm\_for\_task\_5. |
| Функция Algorithm\_for\_task\_5 | Самая длинная функция кода. Принимает целое число в котором будут менять биты и флаг. Флаг это клеймо целого числа, говорит о том, чьи биты в его крови. принимает значения от 1 до 3, где 1 – целое число, 2 – вещественное float, 3 – double  В зависимости от выбора пользователя на какой бит поменять – составляется маска либо полностью состоящая из 0 и 1 единицы, либо полностью из 1 и 1 нуля, что связано с побитовой конъюнкцией и дизъюнкцией. Для каждого флага свой протокол вывода ответа. |
| Функция main | Вызывает функции. Функция - сборка |