**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 И ИХ ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ В ПАМЯТИ**

|  |  |  |
| --- | --- | --- |
| Студент гр. 5372 |  | Уланов С. Е. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2025

**Цель работы.**

Исследование размеров базовых типов данных и их внутреннего двоичного представления в памяти; освоение методов побитового анализа; получение практических навыков работы с различными типами данными и логическими операциями.

**Основные теоретические положения.**

1. **Классификация типов данных**  
   В языке C++ используются различные типы данных:

* Целочисленные: signed char, short int, int, long int, long long int, unsigned char, unsigned short, unsigned int, unsigned long, unsigned long long
* Типы с плавающей точкой: float, double, long double
* Логический: bool
* Символьный тип: char

1. **Оператор sizeof()**  
   Для определения размера типа данных применяется оператор sizeof, возвращающий количество байтов памяти, занимаемых переменной указанного типа.
2. **Представление целых чисел**  
   Целые числа со знаком хранятся в формате дополнительного кода:

* Старший бит является знаковым (0 - положительное число, 1 - отрицательное)
* Диапазон значений: от -2^(n-1) до 2^(n-1)-1, где n - количество битов

1. **Представление типов данных в памяти C++**

**Общие принципы:**

* Каждый тип данных занимает определенное количество байтов в памяти
* Данные хранятся в двоичном виде (последовательности 0 и 1)
* Размеры типов зависят от архитектуры процессора и компилятора Основные типы и их представление:
* **Целые числа:**   
  Хранятся в двоичной системе счисления   
  Знаковые числа используют дополнительный код  
  Старший бит указывает на знак (0 - плюс, 1 - минус)
* **Дробные числа:**   
  Используют стандарт IEEE 754  
  Состоят из трех частей: знак, порядок (экспонента) и мантисса   
  float - 32 бита, double - 64 бита
* **Символы и логические значения:**  
  char хранит код символа (обычно ASCII  
   bool использует 0 для false и не-0 для true

1. **Битовые операции**  
   При работе с битами используются:

* Побитовое И (&)
* Побитовое ИЛИ (|)
* Побитовое исключающее ИЛИ (^)
* НЕ (~)
* Битовые сдвиги (<<, >>)

**Постановка задачи.**

Разработать алгоритм и написать программу, которая позволяет:

1) Вывести, сколько памяти (в байтах) на вашем компьютере отводится под различные типы данных со спецификаторами и без: int, short int, long int, float, double, long double, char и bool.

2) Вывести на экран двоичное представление в памяти (все разряды) целого числа. При выводе необходимо визуально обозначить знаковый разряд и значащие разряды отступами или цветом.

3) Вывести на экран двоичное представление в памяти (все разряды) типа float. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок.

4) Вывести на экран двоичное представление в памяти (все разряды) типа double. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок.

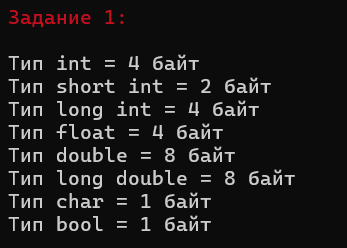
5) Реализовать возможность произвольного изменения любого бита в введенном числе (для всех типов данных) с использованием логических операций.

6) Реализовать операцию умножения числа на 2, используя только битовый сдвиг.

**Выполнение работы.**

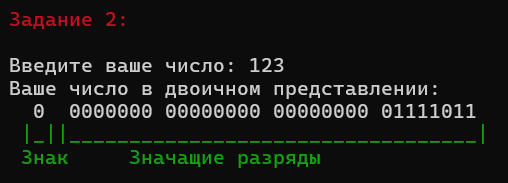
Код программы представлен в приложении А.

**1 задание:**

****

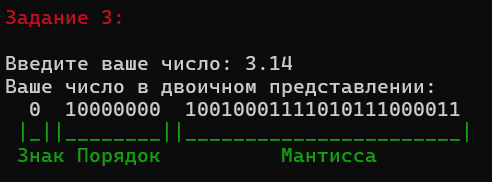
Сразу после запуска программы на экран выводится сколько байт в памяти занимает каждый из типов данных (int, short int, long int, float, double, long double, char и bool).

**2 задание:**



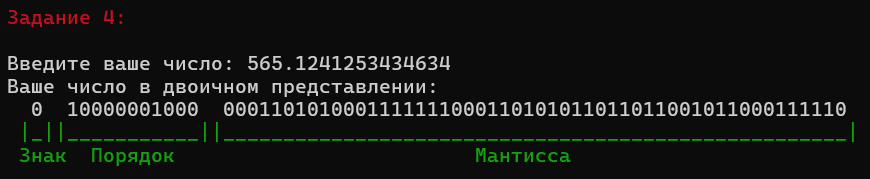
Дальше программа запрашивает у пользователя число (типа int) и выводит его двоичное представление в памяти. Визуально обозначены знаковый бит и значащие разряды.

**3 задание:**



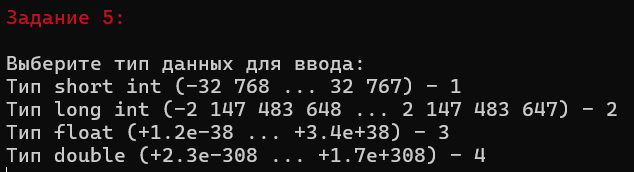
Программа запрашивает у пользователя число (типа float) и выводит его двоичное представление в памяти. Визуально обозначены знаковый бит, порядок и мантисса.

**4 задание:**

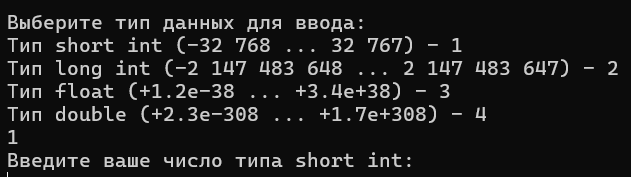


Программа запрашивает у пользователя число (типа double) и выводит его двоичное представление в памяти. Визуально обозначены знаковый бит, порядок и мантисса.

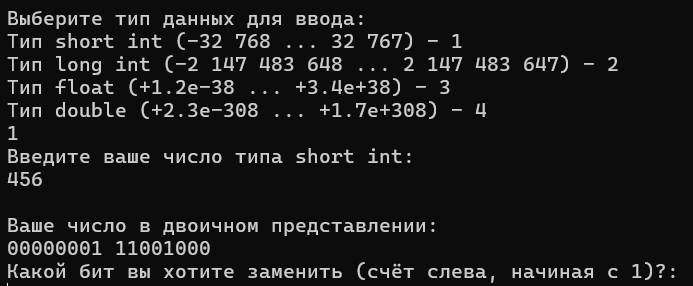
**5 задание:**



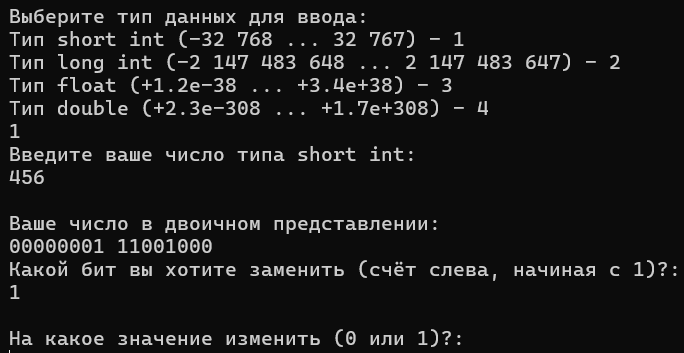
Программа выводит на экран 4 типа данных на выбор (short int, long int, float, double). Пользователь должен выбрать один из типов данных для ввода.



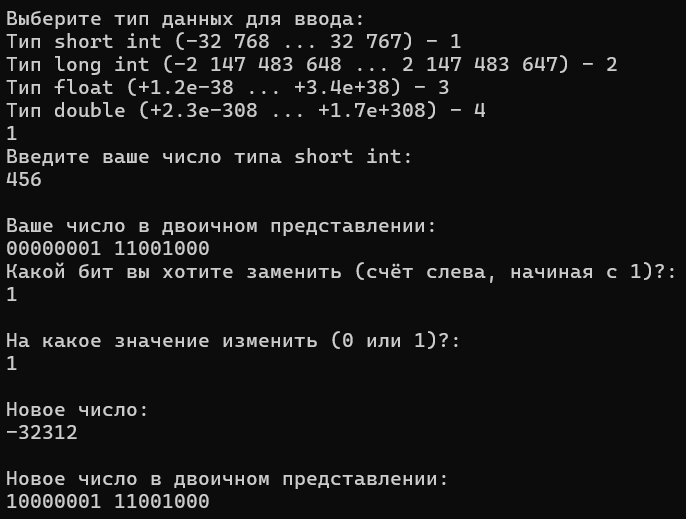
Далее пользователь должен ввести своё число.



После получения числа программа выводит его двоичное представление и предлагает пользователю выбрать бит (счёт начинается слева с 1) для замены.

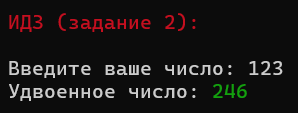


Программа запрашивает значение (0 или 1), на которое необходимо изменить бит.



В конце программа выводит получившееся после замены бита число и его двоичное представление в памяти.

**6. ИДЗ (номер 2):**



Программа запрашивает у пользователя целое число и после получения выводит его удвоенное значение (умножение на 2).

**Выводы.**

В ходе лабораторной работы были изучены принципы внутреннего представления данных в памяти. Практически исследованы размеры основных типов данных и их двоичное представление, включая форматы целых чисел и чисел с плавающей точкой. Освоены методы работы с битовыми операциями: реализовано изменение отдельных битов и сдвиги. Был получен опыт работы с масками. Работа показала практическое применение битовых операций.

Приложение А

рабочий код

