**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(ка) гр. |  |  |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2021

**Цель работы.**

Изучение и организация структур; получение практических навыков работы с типами данных и их внутренне представление в памяти;

**Основные теоретические положения.**

В С++ переменные используются для хранения различных типов данных в памяти компьютера. Каждый тип данных имеет свое собственное выделение памяти.

**Основные 4 спецификатора типа уточняющих внутреннее представление и диапазон значений стандартных типов:**

1. **short**(короткий);
2. **long** (длинный);
3. **signed** (знаковый);
4. **unsigned** (без знаковый).

**Основные типы данных и их размеры:**

1. int: Тип данных, который используется для хранения целых чисел. Занимает 4 байта памяти, диапазон значений от -2,147,483,648 до 2,147,483,647.
2. short int: Целочисленный тип данных, который занимает 2 байта памяти, диапазон значений от -32,768 до 32,767.
3. long int: Целочисленный тип данных для больших чисел, обычно занимает 4 или 8 байтов (в зависимости от компилятора).
4. long long int: Целочисленный тип для очень больших чисел, занимает 8 байтов и может хранить значения от -9,223,372,036,854,775,808 до 9,223,372,036,854,775,807.
5. float: Тип данных с плавающей точкой одинарной точности, занимает 4 байта.
6. double: Тип данных с плавающей точкой двойной точности, занимает 8 байтов и обеспечивает более высокую точность, чем float.
7. long double: Тип данных с плавающей точкой расширенной точности. Занимает 8 или 16 байтов, в зависимости от компилятора.
8. char: Тип данных для хранения символов. Занимает 1 байт и хранит значения от -128 до 127 или от 0 до 255 (для беззнаковых char).
9. bool: Тип данных для логических значений true или false. Занимает 1 байт.

Функция sizeof в C++ используется для определения объема памяти, который занимает переменная или тип данных.

**Представление типов данных в двоичном виде**

1. Тип int занимает 4 байта (32 бита). Старший (самый крайний левый) бит указывает знак числа. Если старший бит равен 0, число положительное, если 1, то отрицательное. Следующие 31 бит обозначают диапазон. Пример числа 5 в двоичном виде: 0 00000000 00000000 00000000 0000101 Пример числа -5 в двоичном виде: 1 11111111 11111111 11111111 1111011
2. Тип float занимает 4 байта (32 бита). Старший (самый крайний левый) бит указывает знак числа. Экспонента: 8 бит - хранит показатель степени с базой 2, смещенный на 127. Мантисса: 23 бита - хранит дробную часть числа в нормализованной форме. Пример числа 5.5 в двоичном виде: 0 10000001 01100000000000000000000
3. Тип double занимает 8 байт (64 бита). Старший (самый крайний левый) бит указывает знак числа. Экспонента: 8 бит - хранит показатель степени с базой 2, смещенный на 1023. Мантисса: 52 бита - хранит дробную часть числа в нормализованной форме. Пример числа 5.5 в двоичном виде: 0 10000000001 0110000000000000000000000000000000000000000000000000

**Постановка задачи.**

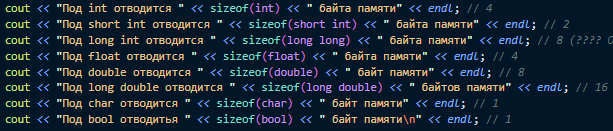
 Вывести, сколько памяти (в байтах) на компьютере отводится под различные типы данных со спецификаторами и без: int, short int, long int, float, double, long double, char и bool.

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

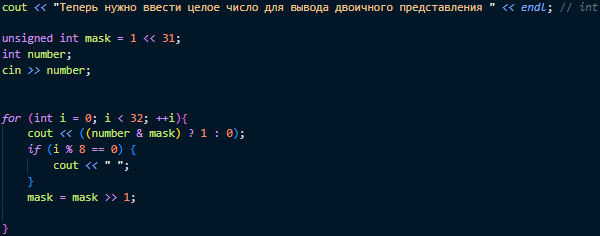
Вывести на экран двоичное представление в памяти (все разряды) типа float и double. Визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок

**Выполнение работы.**

Код программы представлен на гитхабе.

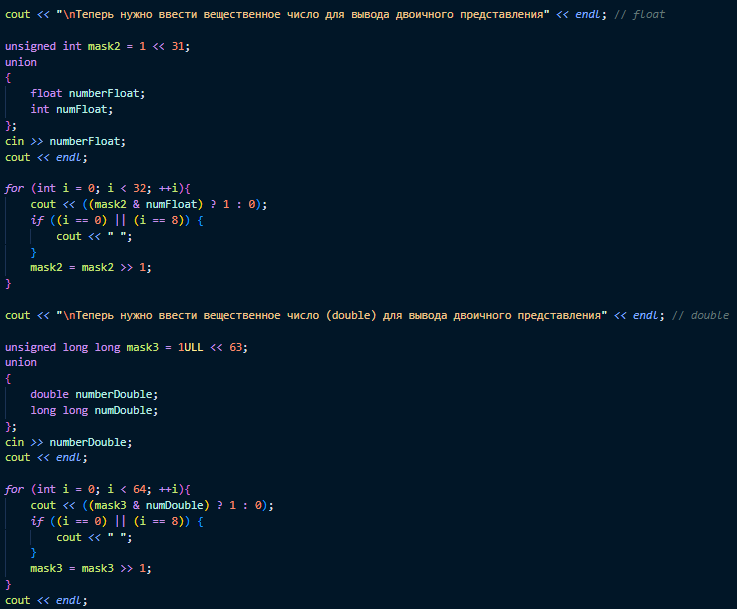


Первый блок кода представляет из себя вывод, сколько в памяти (в байтах) отводится под разные типы данных.



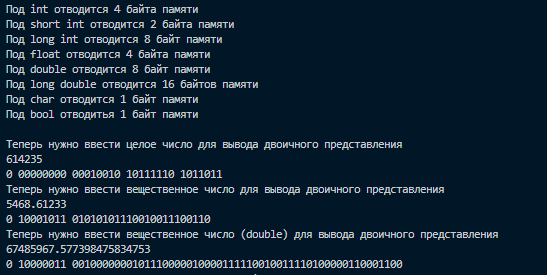
Второй блок кода выводит двоичное представление в памяти (в битах) для типа данных int.

Инициализируется переменная mask со спецификатором unsigned, чтобы он был со старшим знаком и не был отрицательным. Также переменная int, который вводит пользователь. Дальше циклом проходит по каждому биту числа и выполняет операцию побитового умножения. Дальше каждый бит будет выведен на экран, визуально обозначив знаковый разряд и значащие разряды отступами.



Третий блок кода выводит двоичное представление в памяти типов float и double. Инициализируется переменная маски, а также требуемое число. Union используется для того, чтобы получить доступ к одному и тому же значению как к типу float и к типу int (для float), как к типу double и к типу long int (для double). Дальше каждый бит будет выведен на экран, визуально обозначив знаковый разряд, экспоненту и мантиссу отступами.

**Пример работы программы**:



**Выводы.**

В ходе работы было освоено, как различные типы данных хранятся в памяти. Каждый из этих типов имеет собственное двоичное представление. Был использован побитовый сдвиг и побитовые операции для поочередного вывода каждого бита чисел в двоичном формате в памяти.