МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Название кафедры

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Основы алгоритмизации и программирования на языке C++»

Тема: Типы данных и их внутреннее представление в памяти

Студент гр. 1324 _____ Скопцов В.В.

Преподаватель _____ Глущенко А.Г.

Санкт-Петербург

2021

Цель работы.

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.

Основные теоретические положения.

В знаковом целом типе могут быть представлены положительные и отрицательные числа, а также ноль. Положительные числа и ноль в знаковом типе хранятся точно так же, как в беззнаковом – в двоичном виде. Знаковые отрицательные числа хранятся в компьютере в дополнительном коде. Рассмотрим следующий алгоритм.

Обозначим n – разрядность используемого типа данных, X – отрицательное число.

1. Найдем разность: $y=2^n-|X|$

2. Переведем y в двоичную систему счисления.

Введем одно важное правило: если число представлено в памяти в знаковом типе, то старший (крайний левый) бит отвечает за знак числа: если старший бит равен 0, то число положительное, если 1, то отрицательное. Поэтому в знаковых типах старший бит называется Signum (лат. Знак).

Для представления вещественных чисел в памяти компьютера разработан стандарт IEEE 754-2008 1 , в котором описаны форматы представления чисел, алгоритмы округления, операции над числами, обработка исключений (деление на ноль, переполнение и др.). В этом стандарте используется форма представления вещественных чисел с плавающей запятой. В этой форме вещественное число X записывается в следующем виде: $X = \pm m \cdot q^{\pm p}$ где m – мантисса числа (лат. добавление, прибавка), q – основание системы счисления, p – порядок. Для определенности вещественные числа в компьютерах представляются в нормализованном виде, т. е. запятая расположена справа от первой ненулевой цифры.

В двоичном нормализованном представлении числа старший бит, расположенный слева от запятой, всегда равен 1, поэтому его можно не хранить (т. н. «скрытая единица»), а диапазон нормализованных мантисс получается следующим: $1 \leq m < 2$.

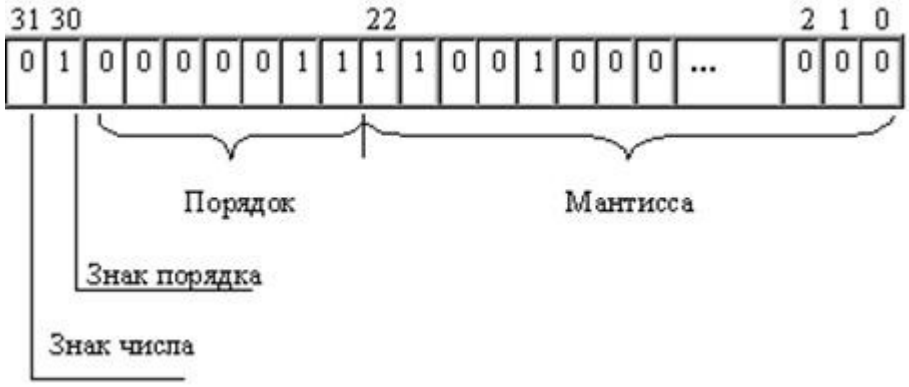
Рассмотрим основные форматы представления вещественных чисел, описанные в стандарте IEEE 754-2008. Они отличаются размером ячейки памяти, отводимой для одного числа:

1) формат одинарной точности (single precision, 4 байт);

2) формат двойной точности (double precision, 8 байт);

3) формат расширенной точности (extended precision, 10 байт).

В формате одинарной точности под число отводится 32 бита, которые распределены следующим образом:



В этом формате указывается только знак всего числа: если бит знака равен 0, число положительное, если 1 – отрицательное. Знак порядка не указывается, так как применяется смещенный порядок sp, который всегда неотрицателен. Смещенный порядок получается прибавлением к исходному порядку константы 127: $sp = p + 127$

Таким образом, для представления вещественного числа X в формате одинарной точности используется следующая формула (в десятичном виде): $X = \pm 1, m \cdot 2^{sp-127}$

В формат двойной точности: под все число отводится 64 бита (8 байт), из них 1 бит – знак, 11 бит – смещенный порядок, 52 бита – мантисса, смещенный порядок вычисляется по формуле: $sp = p + 1023$

Результаты выполнения.

Примеры выполнения программ:

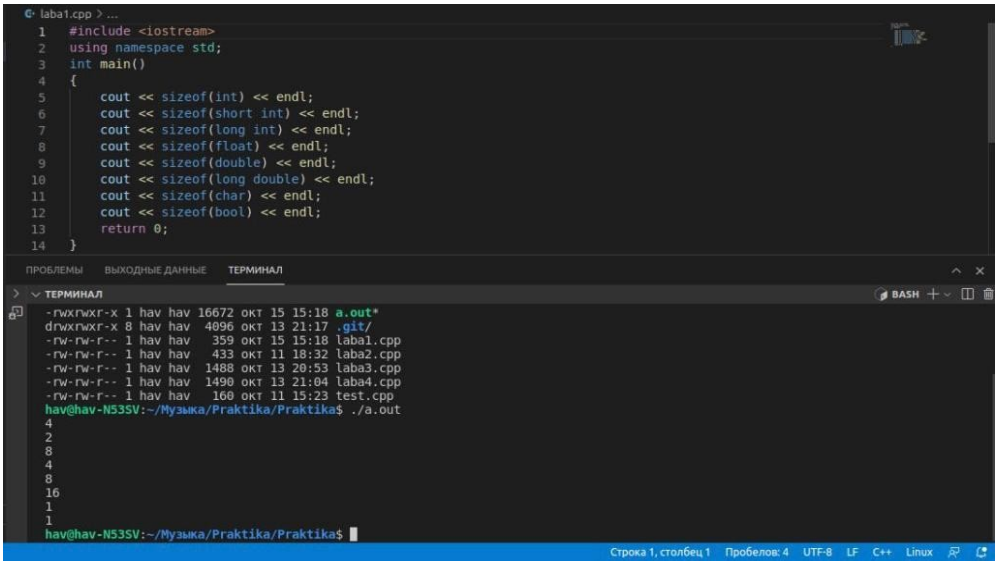


Рисунок 1: Лабораторная работа пункт 1

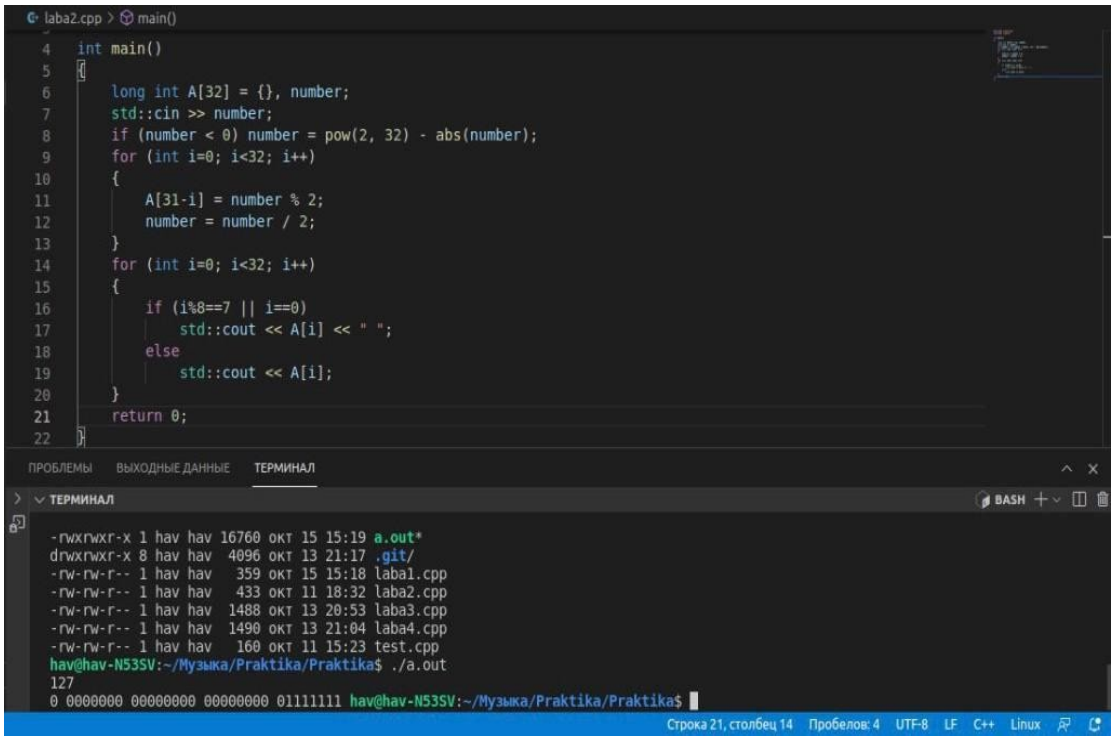


Рисунок 2: Лабораторная работа пункт 2

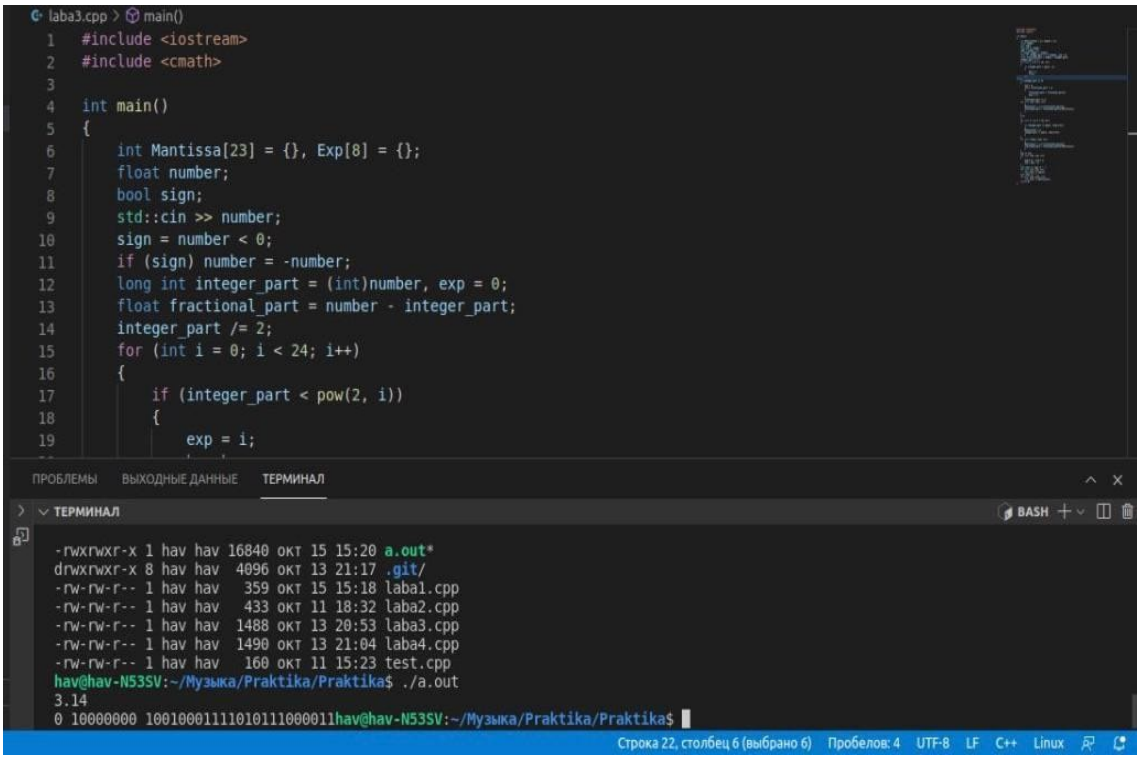


Рисунок 3: Лабораторная работа пункт 3

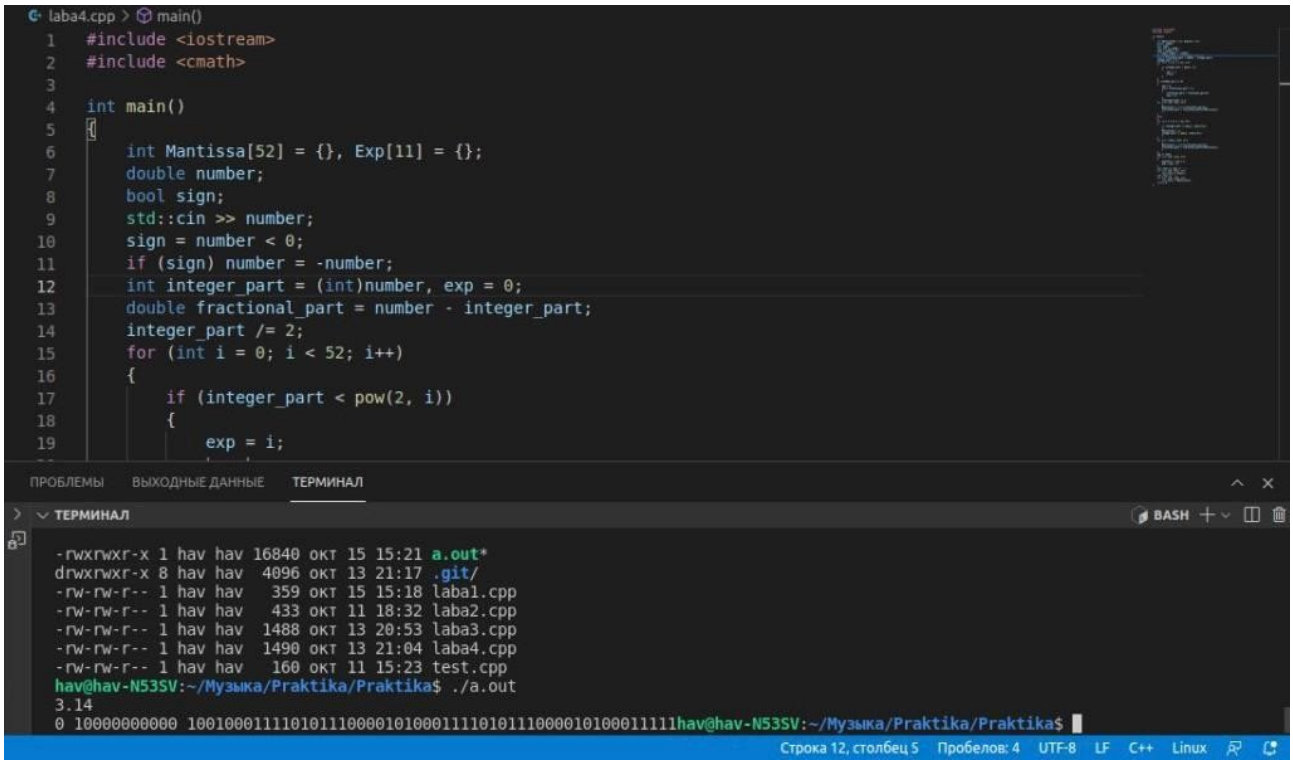


Рисунок 4: Лабораторная работа пункт 4

Выводы.

Написал 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.