

**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**  
**С++»**  
**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.

Внутреннее представление величин целого типа – целое число в двоичном коде. При использовании спецификатора `signed` старший бит числа интерпретируется как знаковый (0 – положительное число, 1 – отрицательное). Для кодирования целых чисел со знаком применяется прямой, обратный и дополнительный коды.

Обратный код получается инвертированием всех цифр двоичного кода абсолютной величины, включая разряд знака: нули заменяются единицами, единицы – нулями. Прямой код можно преобразовать в обратный, инвертировав все значения всех битов (кроме знакового). Обратный код числа  $-3$ :

[illegible]

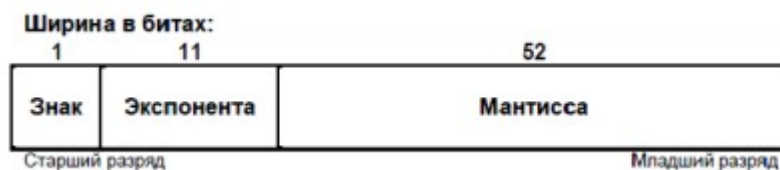
Увидеть, каким образом тип данных представляется на компьютере, можно при помощи логических операций: побитового сдвига ( $\ll$ ) и поразрядной конъюнкции ( $\&$ ).

Putchar возвращает один символ в консоль. Альтернатива - cout. В представленном способе, маска - то, с чем сравнивается значение. И побитовый сдвиг применяется для value. Таким образом 1 бит будет сравниваться с каждым битом числа. Альтернатива - побитовый сдвиг вправо, но при этом нужно проводить данную операцию не над значением(единицей), а над маской (исходным числом, битовое представление которого нужно получить).

При сдвиге вправо для чисел без знака позиции битов, освобожденные при операции сдвига, заполняются нулями. Для чисел со знаком бит знака используется для заполнения освобожденных позиций битов. Другими словами, если число 25 является положительным, используется 0, если число является отрицательным, используется 1. При сдвиге влево позиции битов, освобожденных при операции сдвига, заполняются нулями. Сдвиг влево является логическим сдвигом (биты, сдвигаемые с конца, отбрасываются, включая бит знака).

Вещественные типы данных хранятся в памяти компьютера иначе, чем целочисленные. Внутреннее представление вещественного числа состоит из двух частей – мантиссы и порядка.

Для 32-разрядного процессора для float под мантиссу отводится 23 бита, под экспоненту – 8, под знак – 1. Для double под мантиссу отводится 52 бита, под экспоненту – 11, под знак – 1:



Увидеть, каким образом вещественные типы данных представляются в компьютере немного сложнее. Логические операции, которые использовались с `int`, для вещественных типов данных не подходят. Но это ограничение можно легко обойти, используя объединения.

Объединения – это две или более переменных расположенных по одному адресу (они разделяют одну и ту же память). Объединения определяются с использованием ключевого слова `union`. Объединения не могут хранить одновременно несколько различных значений, они позволяют интерпретировать несколькими различными способами содержимое одной и той же области памяти.

С объединениями нужно быть осторожным. Вся работа с памятью требует грамотного подхода. Более подробно с объединениями можно будет ознакомиться при изучении структур. Пока что объединения будут служить инструментом для работы с `float` и `double`.

### Результаты выполнения.

Примеры выполнения программ:

```
int = 4
short int = 2
long int = 8
float = 4
double = 8
long double = 16
char = 1
bool = 1
hav@hav-N53SV:~/Музыка/Praktika/Praktika$
```

Рисунок 1: Вывод размеров типов данных в байтах.

```
ю №1.pdf'
hav@hav-N53SV:~/Музыка/Praktika/Praktika$ ./a.out
Enter the integer number -127
1 1111111 11111111 11111111 10000001
hav@hav-N53SV:~/Музыка/Praktika/Praktika$
```

Рисунок 2: Вывод представления в памяти числа -127.

```
ю №1.pdf'  
hav@hav-N53SV:~/Музыка/Praktika/Praktika$ ./a.out  
Enter the fractonal number 3.14  
0 10000000 10010001111010111000011  
hav@hav-N53SV:~/Музыка/Praktika/Praktika$
```

Рисунок 3: Вывод представления в памяти числа 3.14 (тип float).

```
-rw-rw-r-- 1 hav hav 546 окт 15 20:10 test.cpp  
-rw-rw-r-- 1 hav hav 352367 окт 15 15:52 'Лабораторная работа по прог  
ю №1.pdf'  
hav@hav-N53SV:~/Музыка/Praktika/Praktika$ ./a.out  
Enter the fractonal number 3.14  
0 100000000000 1001000111101011100001010001111010111000010100011111  
hav@hav-N53SV:~/Музыка/Praktika/Praktika$
```

Рисунок 4: Вывод представления в памяти числа 3.14 (тип double).

## Выводы.

В результате выполнения работы была написанна программа, 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.

Таким образом, программа полностью соответствует условиям поставленной задачи и цели лабораторной работы.