**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. 1324 |  | Мансуров Я. В. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2021

**Цель работы.**

Изучение типов данных и их внутреннего представления в памяти; получение практических навыков работы с типами данных; определение как типы данных представляются на компьютере.

**Основные теоретические положения.**

**Данные** – форма представления информации, с которыми имеют дело информационные системы и их пользователи.

Любая программа предназначена для обработки некоторых данных. Данные представляют некоторую информацию. Информация многообразна – это числовая информация, текстовая информация, аудио и видеоинформация и т. д. Однако, несмотря на многообразие видов информации, внутреннее машинное представление ее едино. Любые данные хранятся в памяти компьютера в виде двоичных кодов.

Типы данных определяет допустимый диапазон изменения переменных и констант, а также допустимые операции над данными этого типа.

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.

Представление положительных и отрицательных чисел в прямом, обратном и дополнительном кодах отличается. В прямом коде в знаковый разряд помещается цифра 1, а в разряды цифровой части числа – двоичный код его абсолютной величины.

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).

Дополнительный код получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду.

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.

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.

**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.

Таблица . Переменные и их назначение

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Тип** | **Назначение** |
| number\_1 | int | Целое число, представление в памяти которого нужно вывести на экран |
| num\_2 | float | Вещественное число(float), представление в памяти которого нужно вывести на экран |
| num\_3 | double | Вещественное число(double), представление в памяти которого нужно вывести на экран |
| a | int | Имеет общую память с number\_2, используется для сдвига |
| j | long long int | Имеет общую память с number\_3, используется для сдвига |
| flag | int | Инициализируется возвращаемым функцией значением для оформления вывода двоичных чисел, её параметр; в зависимости от количества обращений к функции data обеспечивает разделение разрядов разного назначения для вещественных типов |
| mask | unsigned int | Для типов int, float.  Вспомогательная сдвигаемая переменная data, с ней производится сравнение value и с её помощью рассматриваются отдельные биты value |
| unsigned long long int | Для типа double.  Вспомогательная сдвигаемая переменная data, с ней производится сравнение value и с её помощью рассматриваются отдельные биты value |
| order | int | Размер числа в битах, ограничитель цикла в data |

**Ход работы**

1. На экран выводится, сколько памяти (в байтах) на компьютере отводится под различные типы данных со спецификаторами и без в таком порядке:

int, short int, long int, float, double, long double, char, bool.

1. Программа просит ввести целое число и считывает введенное путём функции cin.
2. После определения маски и количества разрядов, программа проходит через цикл for, сравнивая биты и разделяя их

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1. Функция data

1. Объединение в памяти переменных формата float и integer с помощью union.
2. Программа просит ввести вещественное число и считывает введенное.
3. Программа выводит число, записанное в типе данных float.
4. Объединение переменных типов long long integer и double аналогично пункту 4.
5. Программа просит ввести вещественное число и считывает введенное.
6. Программа выводит число, записанное в типе данных double.

**Результат работы программы**

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2. Результат выполнения

Таблица . Ввод и вывод.

|  |  |  |
| --- | --- | --- |
| **Тип данных** | **Ввод** | **Вывод** |
| **integer** | **22** | **0 0000000 00000000 00000000 00010110** |
| **float** | **3.14** | **0 1000000 01001000 11110101 11000011** |
| **double** | **3.14** | **0 10000000000 1001000111101011100001010001111010111000010100011111** |

**Вывод.**

Во время выполнения данной практической работы были изучены типы данных и их внутреннее представление в памяти, были получены практические навыки работы с типами данных, произведена работа с памятью переменных с помощью побитовых операций. Созданная программа способна выводить на экран представление в памяти чисел различного типа.