**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. 2372 |  | Лемешко А. Д. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург, 2022

**Цель работы.**

Понять, как разные типы данных представляются в памяти компьютера.

Написать программу, отображающую представление чисел в памяти компьютера.

**Основные теоретические положения.**

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.

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).

**Постановка задачи.**

Разбить программу на несколько блоков, выполняющих разные функции.

**Первый блок:** вывод количества байт, которое выделяет компьютер на хранение переменных разных типов.

**Второй блок:** вывод двоичного представления числа типа int с визуальным отображением каждого байта и знакового бита в памяти компьютера.

**Третий блок:** вывод двоичного представления числа типа float с визуальным отображением знакового бита, экспоненты и мантиссы в памяти компьютера.

**Четвертый блок:** вывод двоичного представления числа типа double с визуальным отображением знакового бита, экспоненты и мантиссы в памяти компьютера.

**Выполнение работы.**

Код программы представлен в приложении А.

1. При запуске программы пользователь получает объемы памяти, необходимые для хранения чисел в определенных типах.
2. Пользователю предлагается ввести целое число (тип int) для дальнейшего отображения двоичного представления данного числа в памяти компьютера.
3. Пользователю предлагается ввести дробное число (тип float) для дальнейшего отображения двоичного представления данного числа в памяти компьютера.
4. Пользователю предлагается ввести дробное число (тип double) для дальнейшего отображения двоичного представления данного числа в памяти компьютера.

**Выводы.**

**Код написан, и работает корректно.**

Приложение А

рабочий код

#include <iostream>

using namespace std;

int main(){

setlocale(LC\_ALL, "Russian");

/\* Задание 1 - кол-во байтов в памяти у разных типо данных \*/

// вывод размеров типов данных в байтах

cout << "Размер типа int: " << sizeof(int) << "\n"

<< "Размер типа short int: " << sizeof(short int) << "\n"

<< "Размер типа long int: " << sizeof(long int) << "\n"

<< "Размер типа float: " << sizeof(float) << "\n"

<< "Размер типа double: " << sizeof(double) << "\n"

<< "Размер типа long double: " << sizeof(long double) << "\n"

<< "Размер типа char: " << sizeof(char) << "\n"

<< "Размер типа bool: " << sizeof(bool) << endl;

/\* Задание 2 - двоичное представление в памяти int \*/

int numberInt;

cout << "Введите число типа int: ";

cin >> numberInt; // ввод целого числа

int order = 32; // количество разрядов

int mask = 1 << order - 1; // маска побитового сравнения (смещаем влево на 31 бит)

cout << "Ваше число типа int в двоичном виде: ";

for (int i = 1; i <= order; i++){ // вывод двоичного представления

putchar(numberInt & mask ? '1' : '0');

numberInt <<= 1; // Побитовый сдвиг числа

if (i % 8 == 0 || i == 1) // Разделение на байты (8 битов) или отделение знакового бита

{

putchar(' ');

}

}

cout << '\n'; // переход на новую строку

/\* Задание 3 - двоичное представление в памяти float \*/

union{ // объединение переменных

float numberFloat; // значение для представления

int numberIntForFloat; // переменная int, находящаяся в объединении с числом float

};

cout << "Введите число типа float: ";

cin >> numberFloat; // ввод числа типа float

cout << "Ваше число типа float в двоичном виде: ";

// маска для int и float одинаковая

for (int i = 1; i <= order; i++){ // вывод двоичного представления

putchar(numberIntForFloat & mask ? '1' : '0');

numberIntForFloat <<= 1; // Побитовый сдвиг числа

if (i == 9 || i == 1) // Разделение на байты (8 битов) или отделение знакового бита

{

putchar(' ');

}

}

cout << '\n'; // переход на новую строку

/\* Задание 4 - двоичное представление в памяти double \*/

union // объединение int и double

{

double numberDouble; // число типа double

int numberIntForDouble[2]; // переменная int, находящаяся в объединении с числом float (2 штуки, т.к. в один инт double не вместить)

};

cout << "Введите число типа double: ";

cin >> numberDouble;

cout << "Ваше число типа double в двоичном виде: ";

for (int j = 1; j >= 0 ; j--){ // определяет, какой int брать

for (int i = 1; i <= 32; i++){

mask = 1 << order \* 2 - i; // маска для double

putchar(numberIntForDouble[j] & mask ? '1' : '0');

if ((i == 1 && j == 1) || (i == 12 && j == 1)){ // Разделение на байты (8 битов) или отделение знакового бита

putchar(' ');

}

}

}

cout << '\n'; // переход на новую строку

return 0;

}