**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра наименование кафедры**

отчет

**по практической работе №1**

**по дисциплине «Программирование»**

Тема: Типы данных и их внутреннее представление в памяти

|  |  |  |
| --- | --- | --- |
| Студентка гр. 2372 |  | Мельникова М.А. |
| Преподаватель |  | Глущенко А.Г. |

Санкт-Петербург

2022

**Цель работы.**

Ознакомление с типами данных и их внутренним представлением в памяти. Написание программы, выводящей размер памяти, отводимый под разные типы данных и побитовое представление разных типов данных.

**Основные теоретические положения.**

Переменная — это именованная область памяти, в которой хранятся данные определенного типа. Каждая переменная имеет имя и значение.

Инициализация переменной – это первое присвоение ей значения. Все последующие присвоения новых значений этой переменной не считаются инициализацией.

Основные типы данных:

* int (целый) (занимает 2-4 байта памяти);
* float (вещественный) (занимает 4 байта памяти);
* double (вещественный тип с двойной точностью) (занимает 8 байтов памяти);
* bool (логический) (занимает 1 байт памяти);
* char (символьный) (занимает 1 байт памяти).

Спецификаторы типа данных:

* short (короткий);
* long (длинный);
* signed (знаковый);
* unsigned (без знаковый).

Битовая маска — это двоичное число, которое нужно для получения определенных цифр из другого двоичного числа.

Обратный код получается инвертированием всех цифр двоичного кода, включая разряд знака. Прямой код можно преобразовать в обратный, инвертировав все значения битов.

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

Для выведения внутреннего представления чисел различных типов данных используем побитовый сдвиг числа или его маски и поразрядную конъюнкцию.

При сдвиге вправо для положительных чисел освобожденные позиции битов, заполняются нулями. Для отрицательных – единицами. Сдвиг влево является логическим сдвигом (биты, сдвигаемые с конца, отбрасываются, включая бит знака).

Размер типа данных int, зависит от компьютера и компилятора. Для 16-разрядного процессора под величины этого типа отводится 2 байта. Для 32-разрядного - 4 байта и диапазон значений 2 в степени 32.

Данные вещественного типа – это вещественные значения, записанные в памяти в виде чисел с плавающей точкой.

Внутреннее представление вещественного числа состоит из двух частей – мантиссы и порядка (экспоненты). 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.

**Постановка задачи.**

Необходимо 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.

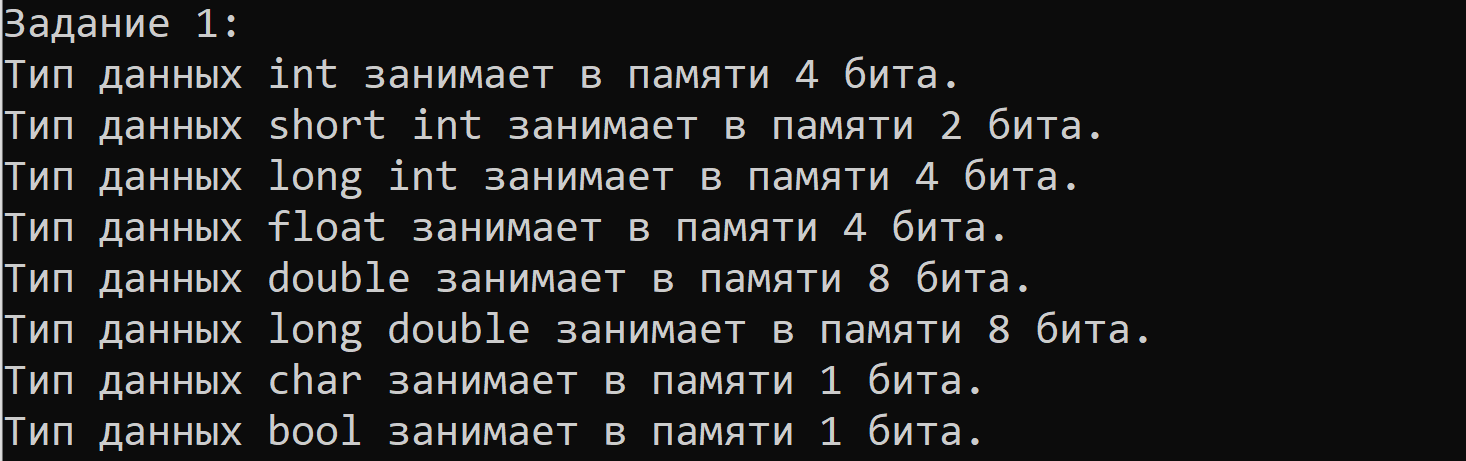
Сделать вывод по полученным результатам и проделанной работе в целом.

**Выполнение работы.**

Для выполнения поставленной задачи работы использован язык программирования С++. Итоговый код программ 1-4 представлен в приложении А.

1. Для выведения количества байт, занимаемых в памяти каждым типом данных, к каждому из типов данных применён оператор sizeof(), показывающий количество байтов памяти, занимаемых указанным типом данных.

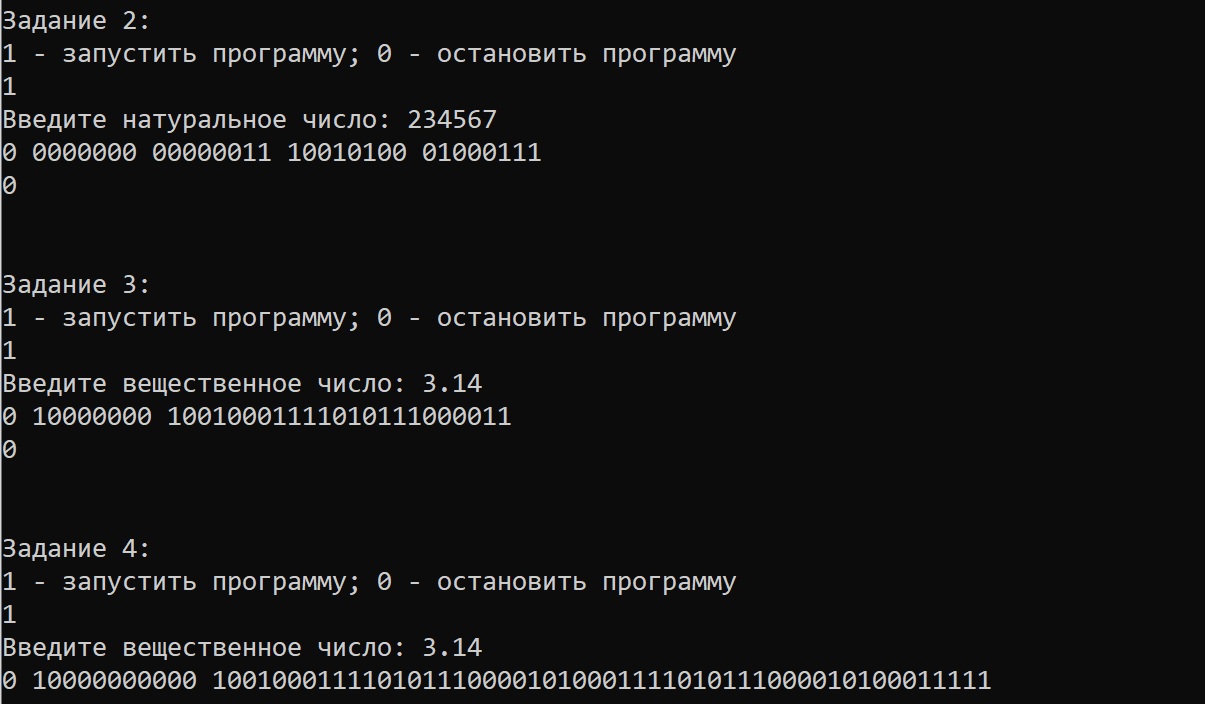
Результат работы программы:



*Рис1*

1. Для выведения двоичного представления числа типа int число вводится с клавиатуры и сохраняется в переменной типа int. Задаётся маска числа mask, равная 231, представляемая как «1» и 31 «0», с которой впоследствии побитово сравнивается исходное число. Задаётся переменная, равная количеству битов в числе типа данных int, равная размеру типа данных, умноженному на количество битов в байте (sizeof(int) \* 8). Впоследствии эта переменная используется в цикле for сравнения со счётчиком цикла. Для наглядности представления информации при выводе каждый бит и первый знаковый символ отделяются пробелами « ». Для этого счётчик цикла for проверяется на кратность 8 и на равенство 1. Для выведения символа «0» или «1» в двоичном представлении введённого числа используется putchar() и побитовая конъюнкция, проверяющая, какое значение бита соответствует единице маски.

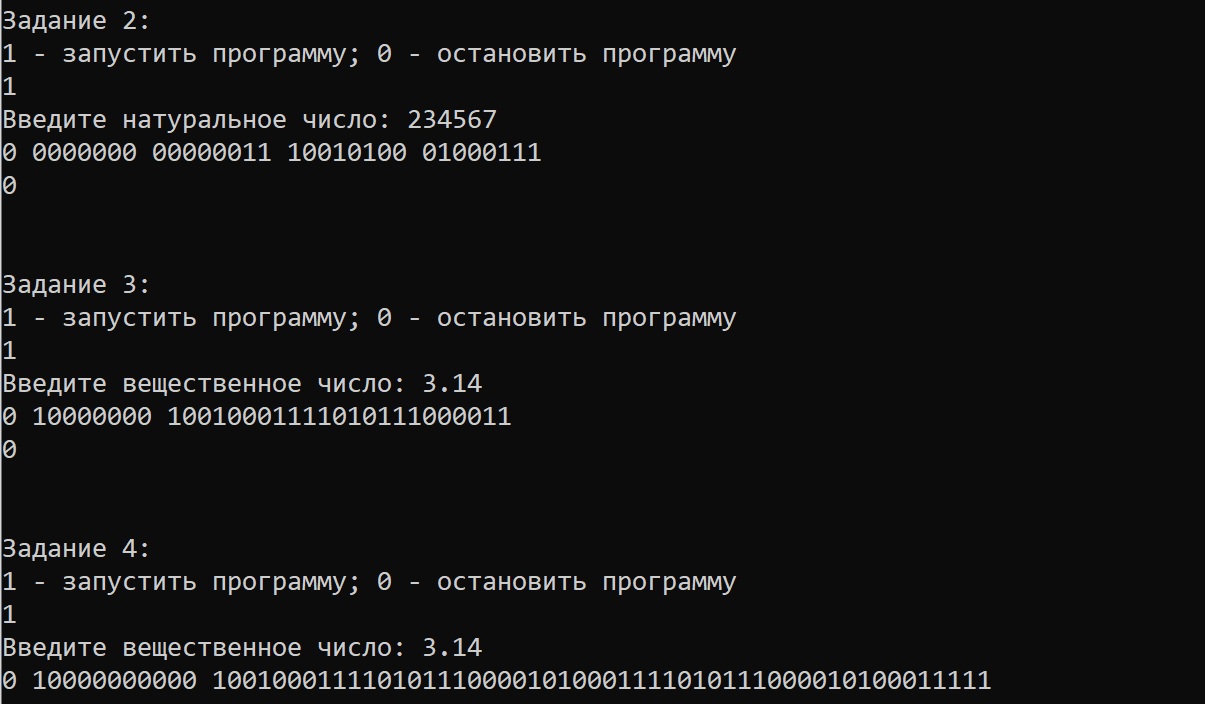
Результат работы программы для произвольного числа:



*Рис2*

1. Для выведения двоичного представления числа типа float используется алгоритм, схожий с алгоритмом программы 2). Только для сравнения числа с маской введённое значения типа float записывается в ячейки памяти, соответствующие типу данных int с помощью объединения union{ }.

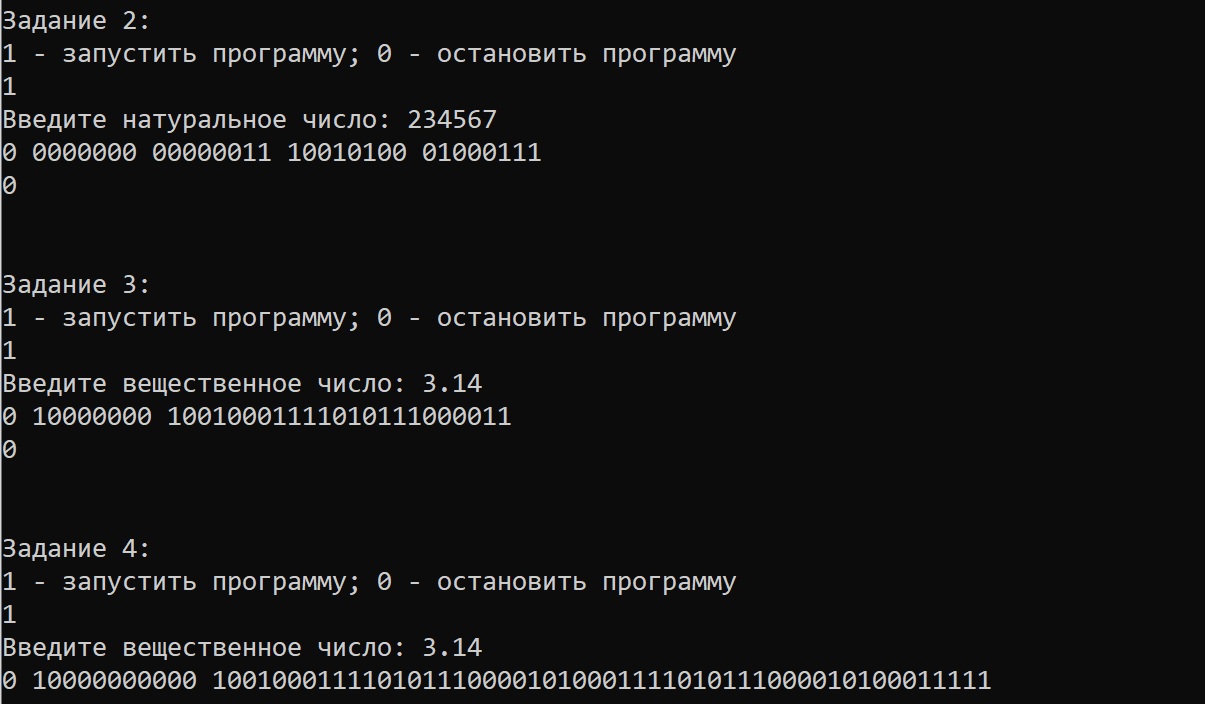
Результат работы программы для произвольного числа:



*Рис3*

1. Для выведения двоичного представления числа типа double используется его объединение с областью памяти, соответствующей массиву из двух элементов int. Тогда с маской сравнивается последовательно первый элемент массива (вторая половина числа), затем нулевой элемент массива (первая половина числа). Для этого используем вложенные циклы.

Результат работы программы для произвольного числа:



*Рис4*

**Выводы.**

В процессе работы было получено представление о типах данных и их внутренним представлением в памяти. Написаны программы, выводящие размер памяти, отводимый под разные типы данных и побитовое представление разных типов данных. Применены операции побитового сдвига, поразрядной конъюнкции.

Приложение А

Полный код программы

#include <iostream>

#include <string>

using namespace std;

int main() {

setlocale(0, "Rus");

cout << "Задание 1:" << "\n"; // Заданеие 1

cout << "Тип данных int занимает в памяти " << sizeof(int) << " бита." << "\n";

cout << "Тип данных short int занимает в памяти " << sizeof(short int) << " бита." << "\n";

cout << "Тип данных long int занимает в памяти " << sizeof(long int) << " бита." << "\n";

cout << "Тип данных float занимает в памяти " << sizeof(float) << " бита." << "\n";

cout << "Тип данных double занимает в памяти " << sizeof(double) << " бита." << "\n";

cout << "Тип данных long double занимает в памяти " << sizeof(long double) << " бита." << "\n";

cout << "Тип данных char занимает в памяти " << sizeof(char) << " бита." << "\n";

cout << "Тип данных bool занимает в памяти " << sizeof(bool) << " бита." << "\n";

cout << "\n" << "\n" << "Задание 2:" << "\n"; // Заданеие 2

cout << "1 - запустить программу; 0 - остановить программу\n";

while (true) {

int n;

cin >> n;

if (n == 1) {

int number;

int order = sizeof(int) \* 8;

unsigned mask = 1 << (order - 1);

cout << "Введите натуральное число: ";

cin >> number;

for (int i = 1; i <= order; ++i)

{

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

mask >>= 1;

if (i == 1 || i % 8 == 0) {

putchar(' ');

}

}

cout << endl;

}

if (n == 0) {

break;

}

if (n != 0 && n != 1)

cout << "Ошибка" << endl;

}

cout << "\n" << "\n" << "Задание 3:" << "\n"; // Заданеие 3

cout << "1 - запустить программу; 0 - остановить программу\n";

while (true) {

int n;

cin >> n;

if (n == 1) {

int order = sizeof(float) \* 8;

unsigned mask = 1 << (order - 1);

union {

int number;

float numberf;

};

cout << "Введите вещественное число: ";

cin >> numberf;

for (int i = 1; i <= order; ++i)

{

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

mask >>= 1;

if (i == 1 || i == 9) {

putchar(' ');

}

}

cout << endl;

}

if (n == 0) {

break;

}

if (n != 0 && n != 1)

cout << "Ошибка" << endl;

}

cout << "\n" << "\n" << "Задание 4:" << "\n"; // Заданеие 4

cout << "1 - запустить программу; 0 - остановить программу\n";

while (true) {

int n;

cin >> n;

if (n == 1) {

union {

int Array[2];

double number;

};

int order = sizeof(double) \* 8;

unsigned mask = 1 << (order - 1);

order = (sizeof(double) \* 8) / 2;

cout << "Введите вещественное число: ";

cin >> number;

int j = 1;

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

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

mask >>= 1;

if (i == 1 || i == 12) {

putchar(' ');

}

}

j = 0;

mask = 1 << (order - 1);

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

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

mask >>= 1;

}

cout << endl;

}

if (n == 0) {

break;

}

if (n != 0 && n != 1)

cout << "Ошибка" << endl;

}

// 13

// Если число отрицательное, то необходимо поменять значение четных битов на 0,

// если положительное, то поменять значение нечетных битов на 1

setlocale(0, "Rus");

cout << "\n" << "\n" << "ИДЗ №13:" << "\n";

cout << "У введённого числа, если оно отрицательное, поменяется значение четныхбитов на 0,\n";

cout << "если оно положительное, поменятся значение нечетных битов на 1\n";

cout << "Выберите тип вводимого числа:\n";

cout << "i - тип дыннх int; f - тип дыннх float; d - тип дыннх double; c - остановить программу\n";

while (true) {

string a;

cin >> a;

if (a == "i") {

cout << "Введите натуральное число: ";

int number;

cin >> number;

int order = sizeof(int) \* 8;

unsigned mask = 1 << (order - 1);

if (number >= 0) {

for (int i = 1; i <= order; i += 2) {

if (!(mask & number)) {

number = mask | number;

}

mask >>= 2;

}

}

else {

mask = 1 << (order - 2);

for (int i = 1; i <= order; i += 2) {

if (mask & number) {

number = ~(mask)&number;

}

mask >>= 2;

}

}

cout << number << endl;

}

else if (a == "f") {

union {

int number;

float number\_f;

};

int order = sizeof(float) \* 8;

unsigned mask = 1 << (order - 1);

cout << "Введите вещественное число: ";

cin >> number\_f;

if (number >= 0) {

for (int i = 1; i <= order; i += 2) {

if (!(mask & number)) {

number = mask | number;

}

mask >>= 2;

}

}

else {

mask = 1 << (order - 2);

for (int i = 1; i <= order; i += 2) {

if (mask & number) {

number = (~mask) & number;

}

mask >>= 2;

}

}

cout << number << endl;

}

else if (a == "d") {

union {

int Array[2];

double number;

};

int order = sizeof(double) \* 8 / 2;

cout << "Введите вещественное число: ";

cin >> number;

if (number >= 0) {

unsigned mask = 1 << (order - 1);

for (int i = 1; i <= order; i += 2) {

if (!(mask & Array[1])) {

Array[1] = mask | Array[1];

}

mask >>= 2;

}

mask = 1 << (order - 1);

for (int i = 1; i <= order; i += 2) {

if (!(mask & Array[0])) {

Array[0] = mask | Array[0];

}

mask >>= 2;

}

}

else {

unsigned mask = 1 << (order - 2);

for (int i = 1; i <= order; i += 2) {

if (mask & Array[1]) {

Array[1] = (~mask) & Array[1];

}

mask >>= 2;

}

mask = 1 << (order - 2);

for (int i = 1; i <= order; i += 2) {

if (mask & Array[0]) {

Array[0] = (~mask) & Array[0];

}

mask >>= 2;

}

}

cout << number << endl;

}

else if (a == "c") {

cout << "Конец" << endl;

break;

}

else {

cout << "Ошибка" << endl;

}

}

}