**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: «Двоичное представление чисел в памяти ЭВМ»

|  |  |  |
| --- | --- | --- |
| Студент гр. 9893 |  | Росихин А.Ю. |
| Преподаватель |  | Глущенко А.Г. |

Санкт-Петербург

2019

**Цель работы.**

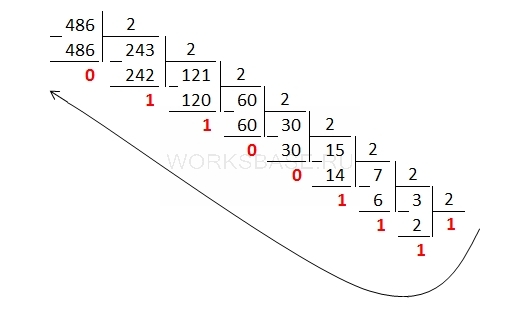
Определить то, как число представляется в памяти ЭВМ. Разобраться, что такое прямой, обратный и дополненный код. Разобраться, что такое порядок, мантисса и нормализованная мантисса. Узнать методы проверки вводимых данный на корректность.

**Основные теоретические положения.**

Перевод десятичного целого и десятичного вещественного числа в двоичное представление.

Целая часть переводится либо методом деления на два либо методом побитового сдвига влево.

Алгоритм перевода целого числа методом деления: выполнить деление исходного числа на 2. Если результат деления больше или равен 2, продолжать делить его на 2 до тех пор, пока результат деления не станет равен 1 [1]. Для примера переведем число 486 в двоичную систему «см. рис. 1»

Рисунок 1 - Перевод целого десятичного числа в двоичную систему

Дополнительный код для отрицательного числа можно получить инвертированием его двоичного модуля (первое дополнение) и прибавлением к инверсии единицы (второе дополнение), либо вычитанием числа из нуля[4].

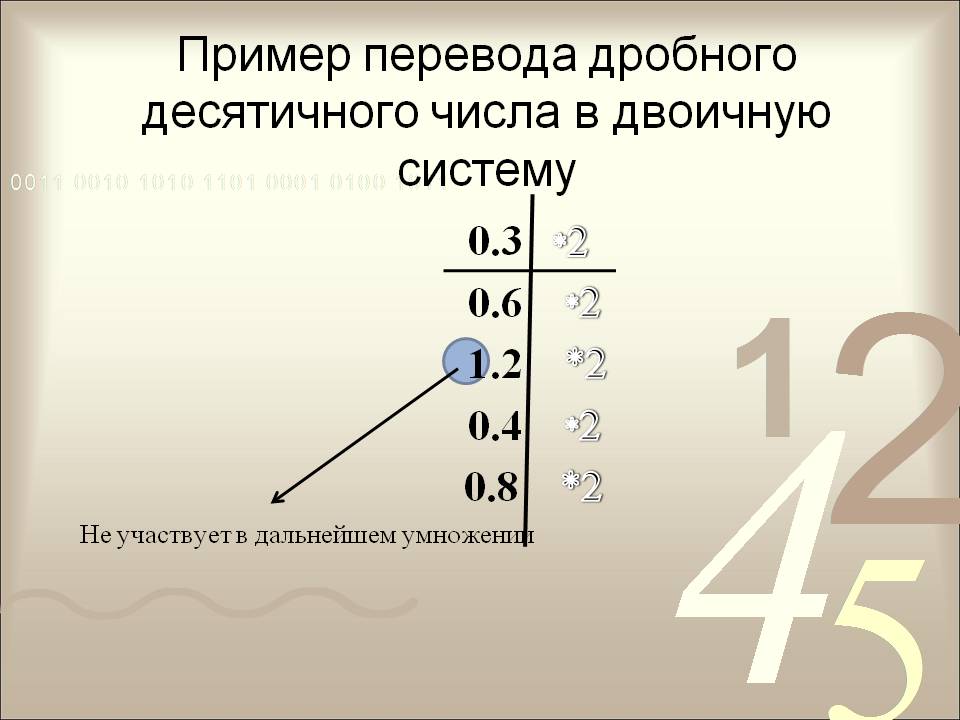
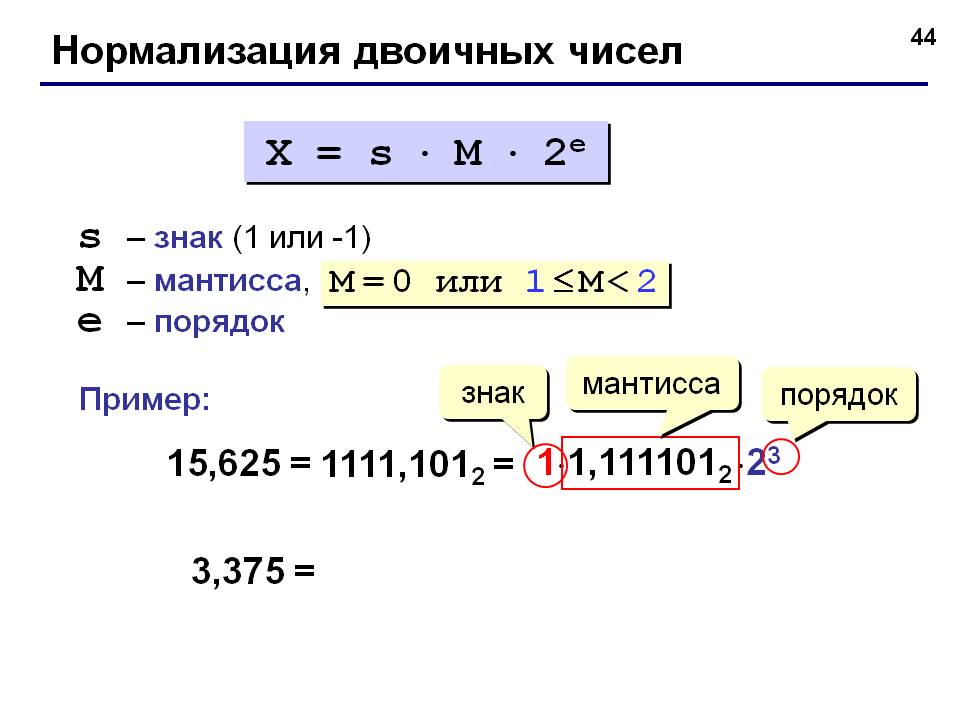
Дробная часть переводится посредством домножения дробной части на два «см. рис. 2» [2]

Рисунок 2 - Перевод дробного десятичного числа в двоичную систему

Перевод числа обратно из двоичного представления в десятичное происходит посредством функции pow() – возведение в степень.

Нормализованный вид мантиссы – так сместить порядок, чтобы число слева от запятой было всегда равно только 1 «см. рис. 3» [3]

Рисунок 3 - Нормализация двоичных чисел

**Постановка задачи.**

1. Разработать алгоритм и написать программу, которая выводит на экран двоичное представление в памяти ЭВМ (все разряды) целого числа (int, shotint, unsignedint). При выводе необходимо визуально обозначить знаковый разряд (отступом) и значащие разряды (цветом);

2. Разработать алгоритм и написать программу, которая выводит на экран двоичное представление в памяти ЭВМ (все разряды) числа типа float, double. При выводе необходимо визуально обозначить знаковый разряд мантиссы, знаковый разряд порядка (если есть), мантиссу и порядок;

3. Написать программа, которая инвертирует все биты в целом числе и числе типа float, double и выводит получившийся результат на экран.

**Выполнение работы.**

Написать программу, которая получает число, введенное с клавиатуры, проверяет корректность ввода, и выводит на экран представление данного числа в памяти ЭВМ в отформатированном виде. Также программа должна выводить инвертированные биты и полученное из этого десятичное число.

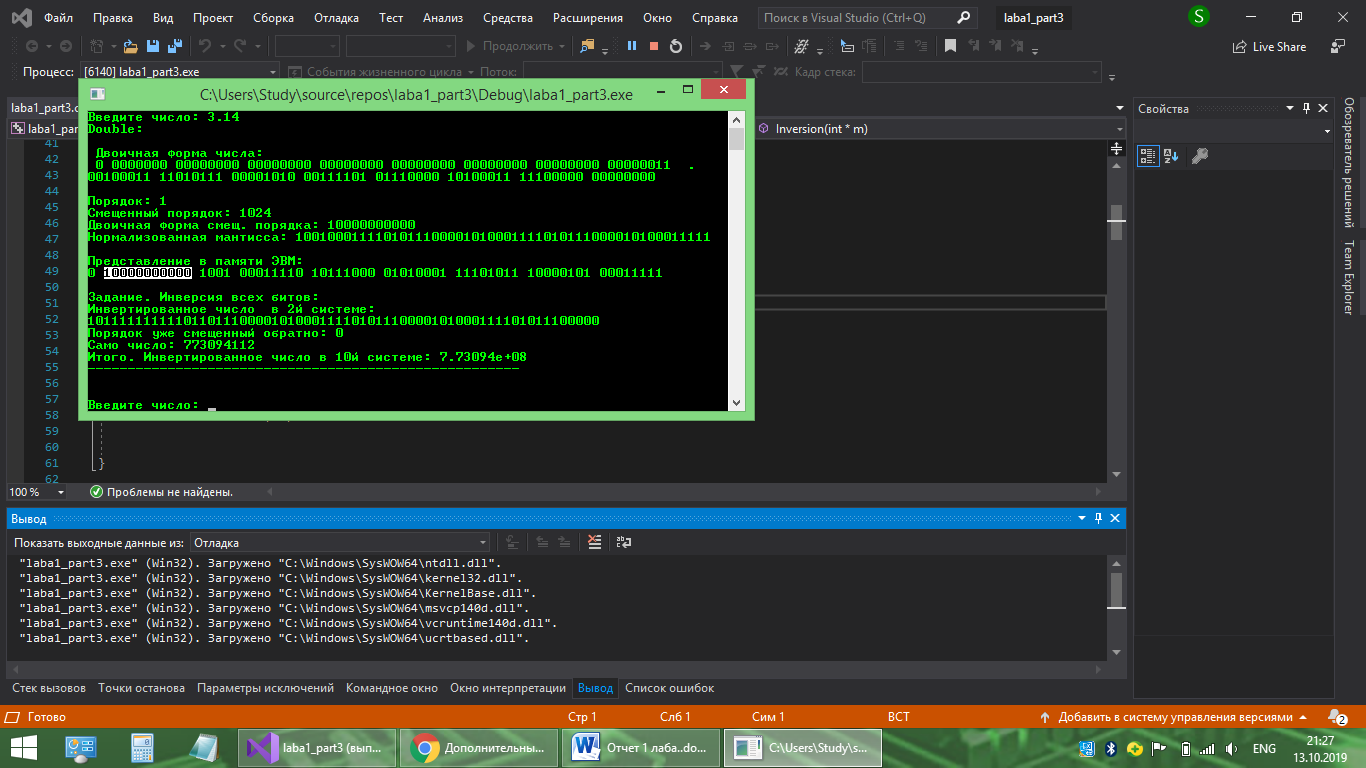
Проверка на «дурака». Проверка, является ли введенное значение числом, если да, то зациклить до бесконечности ввод числа.

Перевод введенного числа в двоичную форму.

Если это целое число и отрицательное число, тогда перевод в обратный и дополнительный код.

Проверка не выходит ли число за границы ShortInt.

Если это вещественное число, тогда перевод отдельно целой и отдельно дробной его части. Вычисление порядка, смещенного порядка.

 Вычисление нормализованной мантиссы. Вывод двоичного представления: знак – порядок – мантисса. Инверсия двоичного кода. Представление инвертированного двоично числа в десятичной форме.

Вывод на экран. «см. рис. 4»

Рисунок 4 - Скриншот работы экрана

Для наглядности привожу пример введенных данных «см. табл. 1»

Таблица 1 – Таблица введенных значений

|  |  |  |
| --- | --- | --- |
| исходное число | двоичная форма числа типа int | двоичная форма числа типа double |
| 3.14 | - | 0 10000000000 1001 00011110 10111000 01010001 11101011 10000101 00011111 |
| 127 | 1 1111111 11111111 11111111 10000001 | 0 10000000101 1111 11000000 00000000 00000000 00000000 00000000 00000000 |
| 10 | 1 1111111 11111111 11111111 11111000 | 0 10000000010 0100 00000000 00000000 00000000 00000000 00000000 00000000 |

**Выводы.**

Для представления целого отрицательно числа в памяти ЭВМ не достаточно просто поменять 1-й бит с 0 на 1. Надо еще перевести из прямого в обратный и дополнительный код.

Дробная часть вещественного числа переводится методом домножения, а не деления, как в случае с целой частью.

Порядок числа должен быть указан со смещением, чтобы он был всегда положительный.

Мантисса должна быть представлена в нормализованном виде и не забыть откинуть 1-ую цифру слева от запятой, т.к. в нормализованном виде она всегда равна 1.

Приложение А

Исходный код программы перевода целого числа в представление ЭВМ

#include<iostream>

#include<string>

usingnamespacestd;

stringperevod(floatchislo)

{

int i = 0, x,b[32], obr[32], dop[32], f[32];

stringstr = "";

x = chislo;

for (int ii = 0; ii < 32; ii++) b[ii] = 0;

i = 0;

while (x != 0)

{b[i] = x % 2;

x = x / 2;

i++; }

str+="Int: "+std::to\_string(b[31]) +" ";

for (int i =30; i >= 24; i--) str+=std::to\_string(b[i]);

str+=" ";

for (int i = 23; i >= 16; i--) str+=std::to\_string(b[i]);

str+=" ";

for (int i = 15; i >= 8; i--) str+=std::to\_string(b[i]);

str+=" ";

for (int i = 7; i >= 0; i--) str+=std::to\_string(b[i]);

str+="\n";

if (chislo> 32767 &&chislo< -32767) str+="ShortInt: NO\n";

else

{str+="ShortInt: "+std::to\_string(b[15]) +" ";

for (int i = 14; i >= 8; i--) str+=std::to\_string(b[i]);

str+=" ";

for (int i = 7; i >= 0; i--) str+=std::to\_string(b[i]);

str+="\n";

}

for (int i3 = 0; i3 < 32; i3++)

if (b[i3] == 0) obr[i3] = 1;

elseobr[i3] = 0;

str+="Обратныйкод: "+std::to\_string(obr[31]) +" ";

for (int i = 30; i >= 24; i--) str+=std::to\_string(obr[i]);

str+=" ";

for (int i = 23; i >= 16; i--) str+=std::to\_string(obr[i]);

str+=" ";

for (int i = 15; i >= 8; i--) str+=std::to\_string(obr[i]);

str+=" ";

for (int i = 7; i >= 0; i--) str+=std::to\_string(obr[i]);

str+="\n";

int n=0;

i = 0;

if (obr[0]!=0) while (n == 0 || i<31) //номер 1гонулевогобита

{

if (obr[i] == 0) n = i;

i++;

}

for (i = 0; i < 32; i++) dop[i] = obr[i];

dop[n] = 1;

if (n>0) for (i = n-1; i >= 0; i--) dop[i] = 0;

str+="Дополнит.код: "+std::to\_string(dop[31]) +" ";

for (int i = 30; i >= 24; i--) str+=std::to\_string(dop[i]);

str+=" ";

for (int i = 23; i >= 16; i--) str+=std::to\_string(dop[i]);

str+=" ";

for (int i = 15; i >= 8; i--) str+=std::to\_string(dop[i]);

str+=" ";

for (int i = 7; i >= 0; i--) str+=std::to\_string(dop[i]);

str+="\n\n\n\n\n\n\n\n\n";

returnstr;

}

int main()

{

setlocale(0, "");

cout<< ("Введите только число: ");

int x;

cin>> x;

cout<<perevod(x);

system("pause");

return 0;

}

**Приложение Б**

**Исходный код программы перевода вещественного числа в представление ЭВМ**

// Лабораторная работа №1 часть 3

// Росихин Александр, гр. 9893

#include<iostream>

#include<string>

#include<cmath>

#include<Windows.h>

usingnamespacestd;

stringshowBinary(int\* m, int\* f) // выводдвоичногопредставлениячисла

{

string s = "";

for (int i = 0; i < 64; i++) // целаячасть

{

s +=to\_string(m[63 - i]);

if (((i + 1) % 8 == 0) || i == 0) s +=" ";

}

s +=" . \n";

for (int i = 0; i < 64; i++) // дробнаячасть

{

s +=to\_string(f[i]);

if ((i + 1) % 8 == 0) s +=" ";

}

return s;

}

voidInversion(int\* m)

{

cout<<"Инвертированное число в 2й системе: "<<endl;

intinversionBinary[64];

for (int i = 0; i < 64; i++)

{

if (m[i] == 0) inversionBinary[i] = 1; elseinversionBinary[i] = 0;

cout<<inversionBinary[i];

}

cout<<endl;

int number = 0;

doubledoubleNumber = 0.0;

int order = -1023;

for (int i = 1; i < 12; i++) order += pow(2, 12 - 1 - i) \* inversionBinary[i];

for (int i = 12; i < 64; i++) number += pow(2, 64 - 1 - i) \* inversionBinary[i];

cout<<"Порядок уже смещенный обратно: "<<order<<endl;

cout<<"Самочисло: "<< number <<endl;

doubleNumber = number \* pow(10, order);

cout<<"Итого. Инвертированное число в 10й системе: "<<doubleNumber;

}

voidGetBinary(doublechislo, int\* intArray, int\* fractionArray)

{

int integer = abs(chislo);

double fraction = abs(chislo) - integer;

for (int i = 0; i < 64; i++) // переводцелойчасти

{

intArray[i] = integer & 1;

integer>>= 1;

}

for (int i = 0; i < 64; i++) // переводдробнойчасти

{

fractionArray[i] = fraction \* 2;

fraction \*= 2;

if (fraction >= 1) fraction -= 1;

}

}

intGetOrder(int\* orderArray, int\* intArray, int\* fractionArray)

{

int order = 0;

for (int i = 63; i > 0; i--) if (intArray[i] != 0) //размерпорядка, есличисло>=1

{

order = i;

break;

}

if (order == 0) for (int i = 0; i < 64; i++) if (fractionArray[i] != 0) // размерпорядка, причисле<1

{

order = -i;

break;

}

cout<<endl<<"Порядок: "<< order <<endl;

intoffsetOrder = 1023 + order;

cout<<"Смещенныйпорядок: "<<offsetOrder<<endl;

for (int i = 0; i < 11; i++) // переводцелойчасти

{

orderArray[10 - i] = offsetOrder& 1;

offsetOrder>>= 1;

}

cout<<"Двоичнаяформасмещ. порядка: ";

for (int i = 0; i < 11; i++) cout<<orderArray[i];

cout<<endl;

return order;

}

voidperevodDouble(doublechislo)

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), (WORD)((0 << 4) | 10));

stringstr = "";

intintegerBinary[64] = {},

fractionBinary[64] = {},

normalizeMantissa[64] = {},

resultBinary[64] = {};

intoffsetOrderBinary[64] = {};

GetBinary(chislo, integerBinary, fractionBinary);

cout<<"Двоичнаяформачисла:\n "+showBinary(integerBinary, fractionBinary) <<endl;

int order = GetOrder(offsetOrderBinary, integerBinary, fractionBinary);

if (order >= 0) for (int i = 1; i <= order; i++) normalizeMantissa[i - 1] = integerBinary[order - i];

if (order >= 0) for (int i = order; i < 52; i++) normalizeMantissa[i] = fractionBinary[i - order];

if (order < 0) for (int i = 0; i < 52; i++) if ((i - order + 1) < 64) normalizeMantissa[i] = fractionBinary[i - order + 1];

cout<<"Нормализованнаямантисса: ";

for (int i = 0; i < 52; i++) cout<<normalizeMantissa[i];

cout<<endl;

if (chislo< 0) resultBinary[0] = 1; elseresultBinary[0] = 0; //знакчисла

for (int i = 1; i < 12; i++) resultBinary[i] = offsetOrderBinary[i - 1];

for (int i = 12; i < 64; i++) resultBinary[i] = normalizeMantissa[i - 12];

cout<<endl<<"Представление в памяти ЭВМ: "<<endl;;

cout<<resultBinary[0] <<" ";

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), (WORD)((15 << 4) | 0));

for (int i = 1; i < 12; i++) cout<<resultBinary[i];

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), (WORD)((0 << 4) | 10));

cout<<" ";

for (int i = 12; i < 16; i++) cout<<resultBinary[i];

cout<<" ";

for (int i = 16; i < 24; i++) cout<<resultBinary[i];

cout<<" ";

for (int i = 24; i < 32; i++) cout<<resultBinary[i];

cout<<" ";

for (int i = 32; i < 40; i++) cout<<resultBinary[i];

cout<<" ";

for (int i = 40; i < 48; i++) cout<<resultBinary[i];

cout<<" ";

for (int i = 48; i < 56; i++) cout<<resultBinary[i];

cout<<" ";

for (int i = 56; i < 64; i++) cout<<resultBinary[i];

cout<<endl<<endl;

cout<<"Задание. Инверсиявсехбитов: "<<endl;

Inversion(resultBinary);

cout<<endl<<"------------------------------------------------------ "<<endl;

}

int main()

{

SetConsoleTextAttribute(GetStdHandle(STD\_OUTPUT\_HANDLE), (WORD)((0 << 4) | 10));

setlocale(0, "");

double x;

while (true)

{

cout<< ("Введите число: ");

cin>> x;

if (cin.fail())

{

cout<<"\nОшибка. Программа будет закрыта.\n";

cin.clear();

break;

}

else

{

cout<<"Double:\n\n ";

perevodDouble(x);

cout<<"\n\n";

}

}

system("pause");

return 0;

}

**Список использованных источников.**

1. <http://worksbase.ru/informatika/teoriya/136-kak-perevesti-chislo.html>
2. <http://900igr.net/prezentacija/matematika/sistemy-schislenija-266653/primer-perevoda-drobnogo-desjatichnogo-chisla-v-dvoichnuju-sistemu-16.html>
3. <http://900igr.net/prezentacija/informatika/kompjuter-iznutri-72960/x-s-44.html>
4. <https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BF%D0%BE%D0%BB%D0%BD%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4>