**Задание 1**

Изучите, как интерпретируется одна и та же область памяти, если она рассматривается как знаковое или беззнаковое число, а также как одно и то же число записывается в различных системах счисления. Необходимо сравнить: а) беззнаковую интерпретацию переменной в шестнадцатеричной форме; б) беззнаковую интерпретацию в десятичной форме; в) знаковую интерпретацию в десятичной форме. Для этого определите и запишите в отчёт десятичное, двоичное (16 бит) и шестнадцатеричное представления шестнадцатибитных чисел 𝑥 и 𝑦, а также беззнаковую интерпретацию этого представления в десятичном виде.

#include <iostream>

#include <cmath>

#include <bitset>

using namespace std;

void print(int Num)

{

cout << "без знаковое шестнадцатеричное: " << hex << (unsigned short)Num << endl << "без знаковое десятичное: " << dec << (unsigned short)Num << endl;

cout << "десятичное: " << dec << (short)Num << endl << "двоичное: " << bitset<16>(Num) << endl;

}

int main()

{

setlocale(LC\_ALL, "Russian");

int x = (-2, 15);

int y = pow(2, 15);

cout << "-2^15: \n";

print(x);

cout << "\n2^15: \n";

print(y);

return 0;

}

-2^15:

без знаковое шестнадцатеричное: f

без знаковое десятичное: 15

десятичное: 15

двоичное: 0000000000001111

2^15:

без знаковое шестнадцатеричное: 8000

без знаковое десятичное: 32768

десятичное: -32768

двоичное: 1000000000000000

**Задание 2**

Найдите и выпишите в отчёт минимальное и максимальное 16- битное число со знаком и без знака в формах представления (а), (б), (в) и в двоичной форме (4 числа, каждое из которых представлено в 4 формах).

#include <iostream>

#include <bitset>

#include <climits>

using namespace std;

void print(int Num)

{

cout << "без знаковое шестнадцатеричное: " << hex << (unsigned short)Num << endl << "без знаковое десятичное: " << dec << (unsigned short)Num << endl;

cout << "десятичное: " << dec << (short)Num << endl << "двоичное: " << bitset<16>(Num) << endl;

}

int main()

{

setlocale(LC\_ALL, "Russian");

cout << "Максимальное 16-битное число без знака: \n\n";

print(USHRT\_MAX);

cout << "\nМинимальное 16-битное число без знака: \n\n";

print(0);

cout << "\nМаксимальное 16-битное число: \n\n";

print(SHRT\_MAX);

cout << "\nМинимальное 16-битное число: \n\n";

print(SHRT\_MIN);

return 0;

}

Максимальное 16-битное число без знака:

без знаковое шестнадцатеричное: ffff

без знаковое десятичное: 65535

десятичное: -1

двоичное: 1111111111111111

Минимальное 16-битное число без знака:

без знаковое шестнадцатеричное: 0

без знаковое десятичное: 0

десятичное: 0

двоичное: 0000000000000000

Максимальное 16-битное число:

без знаковое шестнадцатеричное: 7fff

без знаковое десятичное: 32767

десятичное: 32767

двоичное: 0111111111111111

Минимальное 16-битное число:

без знаковое шестнадцатеричное: 8000

без знаковое десятичное: 32768

десятичное: -32768

двоичное: 1000000000000000

**Задание 3-4.**

Разработайте программу на языке C++, выполняющую над беззнаковыми шестнадцатибитными целыми числами следующие поразрядные операции (результат должен печататься в десятичной и шестнадцатеричной формах):

– бинарные 𝑥 ∧ 𝑦 (конъюнкция), 𝑥 ∨ 𝑦 (дизъюнкция), 𝑥 ⊕ 𝑦 (сложение по модулю два);

– унарные ¬𝑥 (отрицание), neg(𝑥) (дополнение до двух, 𝑥 + neg(𝑥) = 2разрядность 𝑥 );

– 𝑥 ≪ 𝑦 (логический сдвиг влево), 𝑥 ≫ 𝑦 (логический сдвиг вправо).

#include <iostream>

#include <bitset>

#include <cmath>

using namespace std;

template <typename T>

void Show(T x, T y)

{

cout << "Пара 1: " << "Шестнадцатеричное : "<< hex << (T)x << "\t" << "Десятичное : " << dec << (T)x << endl;

cout << "Пара 2: " << "Шестнадцатеричное : "<< hex << (T)y << "\t" << "Десятичное : " << dec << (T)y << endl;

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

}

template <typename T>

T Conjunction(T x,T y)

{

return (x&y);

}

template <typename T>

T Disjunction(T x,T y)

{

return (x|y);

}

template <typename T>

T Not(T x)

{

return ~x;

}

template <typename T>

void print(T x,T y)

{

cout <<"Унарные операции:\n" << "\nОтрицание:\n" << endl;

cout << "~x:" << "\tДесятичное: " << dec << T(~x) << "\tШестнадцатеричное: " << hex << T(~x) << endl;

cout << "neg(x)" << "\tДесятичное: " << dec << T(-x) << "\tШестнадцатеричное: " << hex << T(-x)<< endl;

cout << "\nЛогичские операции:\n" << "\nЛогический сдвиг влево:\n";

cout << "x<<y:" << "\tДесятичное: " << dec << T(x<<y) << "\tШестнадцатеричное: " << hex << T(x<<y) << endl;

cout << "Логический сдвиг вправо:\n";

cout << "x>>y:" << "\tДесятичное: " << dec << T(x>>y) << "\tШестнадцатеричное: " << hex << T(x>>y) << endl << endl;

}

template<typename T>

T Xor(T x, T y)

{

return x^y;

}

int main()

{

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

unsigned short int xu1, yu1, xu2, yu2;

xu1 = 0x8008;

yu1 = 0x1111;

xu2 = 0x0005;

yu2 = 0x0002;

cout << "Бинарные операции: " << endl;

cout <<"И: " << endl;

Show(Conjunction(xu1,yu1),Conjunction(xu2,yu2));

cout <<"ИЛИ: " << endl;

Show(Disjunction(xu1,yu1),Disjunction(xu2,yu2));

cout <<"Sum2: " << endl;

Show(Xor(xu1,yu1),Xor(xu2,yu2));

cout << "Пара 1: " << endl << endl;

print(xu1,yu1);

cout << "Пара 2: " << endl << endl;

print(xu2,yu2);

cout << "---------------Со знаком--------------------" << endl;

signed short int x1, y1, x2, y2;

x1 = 0x8008;

y1 = 0x1111;

x2 = 0x0005;

y2 = 0x0002;

cout << "Бинарные операции: " << endl;

cout <<"И: " << endl;

Show(Conjunction(x1,y1),Conjunction(x2,y2));

cout <<"ИЛИ: " << endl;

Show(Disjunction(x1,y1),Disjunction(x2,y2));

cout <<"Сложение по модулю 2: " << endl;

Show(Xor(x1,y1),Xor(x2,y2));

print(x1,y1);

print(x2,y2);

signed int x = x1;

cout << "short int x:" << "\tДесятичное: " << dec << x1 << "\tШестнадцатеричное: " << hex << x1 << endl << endl;

cout << "int x:" << "\tДесятичное: " << dec << x << "\tШестнадцатеричное: " << hex << x << endl << endl;

return 0;

}

--------------Без знаковые-------------------

Бинарные операции:

И:

Пара 1: Шестнадцатеричное : 0 Десятичное : 0

Пара 2: Шестнадцатеричное : 0 Десятичное : 0

----------

ИЛИ:

Пара 1: Шестнадцатеричное : 9119 Десятичное : 37145

Пара 2: Шестнадцатеричное : 7 Десятичное : 7

----------

Sum2:

Пара 1: Шестнадцатеричное : 9119 Десятичное : 37145

Пара 2: Шестнадцатеричное : 7 Десятичное : 7

----------

Пара 1:

Унарные операции:

Отрицание:

~x: Десятичное: 32759 Шестнадцатеричное: 7ff7

neg(x) Десятичное: 32760 Шестнадцатеричное: 7ff8

Логические операции:

Логический сдвиг влево:

x<<y: Десятичное: 0 Шестнадцатеричное: 0

Логический сдвиг вправо:

x>>y: Десятичное: 0 Шестнадцатеричное: 0

Пара 2:

Унарные операции:

Отрицание:

~x: Десятичное: 65530 Шестнадцатеричное: fffa

neg(x) Десятичное: 65531 Шестнадцатеричное: fffb

Логические операции:

Логический сдвиг влево:

x<<y: Десятичное: 20 Шестнадцатеричное: 14

Логический сдвиг вправо:

x>>y: Десятичное: 1 Шестнадцатеричное: 1

---------------Со знаком--------------------

Бинарные операции:

И:

Пара 1: Шестнадцатеричное: 0 Десятичное: 0

Пара 2: Шестнадцатеричное: 0 Десятичное: 0

----------

ИЛИ:

Пара 1: Шестнадцатеричное: 9119 Десятичное: -28391

Пара 2: Шестнадцатеричное: 7 Десятичное: 7

----------

Сложение по модулю 2:

Пара 1: Шестнадцатеричное: 9119 Десятичное: -28391

Пара 2: Шестнадцатеричное: 7 Десятичное: 7

----------

Унарные операции:

Отрицание:

~x: Десятичное: 32759 Шестнадцатеричное: 7ff7

neg(x) Десятичное: 32760 Шестнадцатеричное: 7ff8

Логические операции:

Логический сдвиг влево:

x<<y: Десятичное: 0 Шестнадцатеричное: 0

Логический сдвиг вправо:

x>>y: Десятичное: -1 Шестнадцатеричное: ffff

Унарные операции:

Отрицание:

~x: Десятичное: -6 Шестнадцатеричное: fffa

neg(x) Десятичное: -5 Шестнадцатеричное: fffb

Логические операции:

Логический сдвиг влево:

x<<y: Десятичное: 20 Шестнадцатеричное: 14

Логический сдвиг вправо:

x>>y: Десятичное: 1 Шестнадцатеричное: 1

short int x: Десятичное: -32760 Шестнадцатеричное: 8008

int x: Десятичное: -32760 Шестнадцатеричное: ffff8008

**Задание 6.**

Определите и выпишите в отчёт, как хранятся в памяти компьютера: – целое число 0x12345678; по результату исследования определите порядок следования байтов в словах для вашего процессора: – Little-Endian (от младшего к старшему, порядок Intel); – Big-Endian (от старшего к младшему, порядок Motorola);

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Win 32 | | Win 64 | | Ubuntu 32 | | Ubuntu 64 | |
| 0x12345678 | 78 56 34 12 | xV4. | 78 56 34 12 | xV4. | 78 56 34 12 | xV4. | 78 56 34 12 | xV4. |
| "abcd" | 61 62 63 64 00 | abcd. | 61 62 63 64 00 | abcd. | 61 62 63 64 00 | abcd. | 61 62 63 64 00 | abcd. |
| "абвг" | e0 e1 e2 e3 00 | абвг. | e0 e1 e2 e3 00 | абвг. | d0 b0 d0 b1 d0 b2 d0 b3 | Ð°Ð±Ð²Ð³ | d0 b0 d0 b1 d0 b2 d0 b3 | Ð°Ð±Ð²Ð³ |
| L"abcd" | 61 00 62 00 63 00 64 00 | a.b.c.d. | 61 00 62 00 63 00 64 00 | a.b.c.d. | 61 00 00 00 62 00 00 00 63 00 00 00 64 00 00 00 | a...b...c...d... | 61 00 00 00 62 00 00 00 63 00 00 00 64 00 00 00 | a...b...c...d... |
| L"абвг" | 30 04 31 04 32 04 33 04 | 0.1.2.3. | 30 04 31 04 32 04 33 04 | 0.1.2.3. | 30 04 00 00 31 04 00 00 32 04 00 00 33 04 00 00 | 0...1...2...3... | 30 04 00 00 31 04 00 00 32 04 00 00 33 04 00 00 | 0...1...2...3... |

**Задание 7.**

При помощи оператора sizeof выясните, сколько байтов занимают переменные следующих типов: char, bool, wchar\_t, short, int, long, long long, float, double, long double, size\_t, ptrdiff\_t, void\*. Результаты оформите в отчёте в виде таблицы, указывая для каждого типа его назначение. Для выполнения единообразных действий над переменными различных типов используются макросы препроцессора C или шаблоны C++.

#include <iostream>

#include <cstddef>

template<typename T>

void printSize()

{

std::cout << sizeof(T) << std::endl;

}

int main()

{

printSize<char>();

printSize<bool>();

printSize<wchar\_t>();

printSize<short>();

printSize<int>();

printSize<long>();

printSize<long long>();

printSize<float>();

printSize<double>();

printSize<long double>();

printSize<size\_t>();

printSize<ptrdiff\_t>();

printSize<void\*>();

return 0;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Тип данных | Длина в байтах | | | | | Назначение |
|  | Win32 | Win64 | Ubuntu 32 | Ubuntu 64 | Стандарт С++14 |  |
| char | 1 | 1 | 1 | 1 | 1 | целочисленный (символьный) тип данных |
| bool | 1 | 1 | 1 | 1 | 1 | целочисленный (логический) тип данных |
| wchar\_t | 2 | 2 | 4 | 4 | Не меньше, чем 2 | unicode-тип символов |
| short | 2 | 2 | 2 | 2 | не меньше чем 2 | целочисленный тип, размер которого больше или равен размеру типа char и меньше или равен размеру типа int |
| int | 4 | 4 | 4 | 4 | не меньше чем 2 | int — это целочисленный тип, размер которого больше или равен размеру типа short int и меньше или равен размеру типа long |
| long | 4 | 4 | 4 | 8 | не меньше чем 4 | long (или long int) — это целочисленный тип, размер которого больше или равен размеру типа int |
| long long | 8 | 8 | 8 | 8 | не меньше чем 8 | Больше, чем unsigned long |
| float | 4 | 4 | 4 | 4 | 4 | float — это тип с плавающей запятой наименьшего размера. |
| double | 8 | 8 | 8 | 8 | Не меньше, чем 4 | double — это тип с плавающей запятой, размер которого больше или равен размеру типа float, но меньше или равен размеру типа long double |
| long double | 8 | 8 | 12 | 16 | не меньше чем 8 | long double — это тип с плавающей запятой, размер которого больше или равен размеру типа double |
| size\_t | 4 | 8 | 4 | 8 | не меньше чем 2 | целочисленный беззнаковый тип результата, возвращаемого операторами [sizeof](http://ru.cppreference.com/w/cpp/language/sizeof) и [alignof](http://ru.cppreference.com/w/cpp/language/alignof) |
| ptrdiff\_t | 4 | 8 | 4 | 8 | не меньше чем 4 | базовый знаковый целочисленный тип, является типом результата выражения, где один указатель вычитается из другого |
| void\* | 4 | 8 | 4 | 8 | не меньше чем 4 | Указатель на неопределенный тип,  в этот указатель мы можем поместить любой тип. |