**Задание 1**

**void print16(void \*p)**

**{**

**unsigned short \*a=reinterpret\_cast<unsigned short\*>(p);**

**bitset<16> bitform(\*a);**

**short \*b=reinterpret\_cast<short\*>(p);**

**cout<<setfill('0');**

**cout<<setw(4)<<hex<<\*a<<'/'<<setw(16)<<bitform<<'/'<<setfill(' ')<<setw(5)<<dec<<\*a<<'/'<<setfill(' ')<<showpos<<setw(6)**

**<<\*b<<'/'<<endl<<endl;**

**}**

**void print32(void \*p)**

**{**

**unsigned int \*i = reinterpret\_cast<unsigned int \*>(p);**

**bitset<32> bitform(\*i);**

**cout << setfill('0')<< setw(8) << hex << \*i << '/'<< setw(32) << bitform << '/'<< setfill(' ') << setw(10) << dec << \*i << '/';**

**int \*c = reinterpret\_cast<int \*>(p);**

**cout << setfill(' ') << showpos << setw(11) << \*c << '/';**

**float \*b = reinterpret\_cast<float \*>(p);**

**cout << setfill(' ') << setw(55) << dec << setprecision(50) << fixed << \*b**

**<< '/';**

**cout << setfill(' ') << setw(10) << dec << setprecision(3) << scientific**

**<< \*b << endl**

**<< endl;**

**}**

**short a=2;**

**print16(&a);**

**int b=3;**

**print32(&b);**

**0002/0000000000000010/ 2/ +2/**

**00000003/00000000000000000000000000000011/ 3/ +3/ +0.00000000000000000000000000000000000000000000420390/+4.204e-45**

**Задание 2**

**cout<<"\n\nЗадание 2\_\_\_\_\_\_\_\_\_\_"<<endl;**

**unsigned short m1 = 986;**

**unsigned int n1 = static\_cast<unsigned int>(m1);**

**print16(&m1);**

**print32(&n1);**

**cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;**

**signed short m2 = 986;**

**signed int n2 = static\_cast<signed int>(m2);**

**print16(&m2);**

**print32(&n2);**

**cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;**

**unsigned short m3 = -126;**

**unsigned int n3 = static\_cast<unsigned int>(m3);**

**print16(&m3);**

**print32(&n3);**

**cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;**

**signed short m4 = -126;**

**signed int n4 = static\_cast<signed int>(m4);**

**print16(&m4);**

**print32(&n4);**

**cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;**

03da/0000001111011010/ 986/ +986/

000003da/00000000000000000000001111011010/ 986/ +986/ +0.00000000000000000000000000000000000000000138168029/+1.382e-42

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

03da/0000001111011010/ 986/ +986/

000003da/00000000000000000000001111011010/ 986/ +986/ +0.00000000000000000000000000000000000000000138168029/+1.382e-42

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ff82/1111111110000010/65410/ -126/

0000ff82/00000000000000001111111110000010/ 65410/ +65410/ +0.00000000000000000000000000000000000000009165893255/+9.166e-41

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ff82/1111111110000010/65410/ -126/

ffffff82/11111111111111111111111110000010/4294967170/ -126/

+nan/ +nan

**Задание 3**

**cout<<"\n\nЗадание 3\_\_\_\_\_\_\_\_\_\_"<<endl;**

**signed short m1 = 986;**

**unsigned short m2 = 986;**

**signed short n1 = -126;**

**unsigned short n2 = -126;**

**signed int count1 = 2;**

**signed int pr1 = m1 \* count1;**

**cout << endl**

**<< "Знаковое умножение m и n на 2, значение и результат для числа m и "**

**"числа n:"**

**<< endl;**

**cout << "m:" << endl;**

**print16(&m1);**

**cout << "result:" << endl;**

**print16(&pr1);**

**pr1 = n1 \* count1;**

**cout << "n:" << endl;**

**print16(&n1);**

**cout << "result:" << endl;**

**print16(&pr1);**

**unsigned int count2 = 2;**

**unsigned int pr2 = m2 \* count2;**

**cout << endl**

**<< "Беззнаковое умножение m и n на 2, значение и результат для числа "**

**"m и числа n:"**

**<< endl;**

**cout << "m:" << endl;**

**print16(&m2);**

**cout << "result:" << endl;**

**print16(&pr2);**

**pr2 = n2 \* count2;**

**cout << "n:" << endl;**

**print16(&n2);**

**cout << "result:" << endl;**

**print16(&pr2);**

**signed int del1 = m1 / count1;**

**cout << endl**

**<< "Знаковое деление m и n на 2, значение и результат для числа m и "**

**"числа n:"**

**<< endl;**

**cout << "m:" << endl;**

**print16(&m1);**

**cout << "result:" << endl;**

**print16(&del1);**

**del1 = n1 / count1;**

**cout << "n:" << endl;**

**print16(&n1);**

**cout << "result:" << endl;**

**print16(&del1);**

**unsigned int del2 = m2 / count2;**

**cout << endl**

**<< "Беззнаковое деление m и n на 2, значение и результат для числа m "**

**"и числа n:"**

**<< endl;**

**cout << "m:" << endl;**

**print16(&m2);**

**cout << "result:" << endl;**

**print16(&del2);**

**del2 = n2 / count2;**

**cout << "n:" << endl;**

**print16(&n2);**

**cout << "result:" << endl;**

**print16(&del2);**

**count2 = 16;**

**unsigned int ost = m2 % count2;**

**cout << endl<< "Беззнаковое деление m и n на 16, значение и результат для числа m "**

**"и числа n:"<< endl;**

**cout << "m:" << endl;**

**print16(&m2);**

**cout << "result:" << endl;**

**print16(&ost);**

**ost = n2 % count2;**

**cout << "n:" << endl;**

**print16(&n2);**

**cout << "result:" << endl;**

**print16(&ost);**

**unsigned int round = m2 - (m2 % count2);**

**cout << endl << "Округление вниз до числа, кратного 16" << endl;**

**cout << "m:" << endl;**

**print16(&m2);**

**cout << "result:" << endl;**

**print16(&round);**

**ost = n2 - (n2 % count2);**

**cout << "n:" << endl;**

**print16(&n2);**

**cout << "result:" << endl;**

**print16(&round);**

Знаковое умножение m и n на 2, значение и результат для числа m и числа n:

m:

03da/0000001111011010/ 986/ +986/

result:

07b4/0000011110110100/ 1972/ +1972/

n:

ff82/1111111110000010/65410/ -126/

result:

ff04/1111111100000100/65284/ -252/

Беззнаковое умножение m и n на 2, значение и результат для числа m и числа n:

m:

03da/0000001111011010/ 986/ +986/

result:

07b4/0000011110110100/ 1972/ +1972/

n:

ff82/1111111110000010/65410/ -126/

result:

ff04/1111111100000100/65284/ -252/

Знаковое деление m и n на 2, значение и результат для числа m и числа n:

m:

03da/0000001111011010/ 986/ +986/

result:

01ed/0000000111101101/ 493/ +493/

n:

ff82/1111111110000010/65410/ -126/

result:

ffc1/1111111111000001/65473/ -63/

Беззнаковое деление m и n на 2, значение и результат для числа m и числа n:

m:

03da/0000001111011010/ 986/ +986/

result:

01ed/0000000111101101/ 493/ +493/

n:

ff82/1111111110000010/65410/ -126/

result:

7fc1/0111111111000001/32705/+32705/

Беззнаковое деление m и n на 16, значение и результат для числа m и числа n:

m:

000a/0000000000001010/ 10/ +10/

n:

ff82/1111111110000010/65410/ -126/

result:

0002/0000000000000010/ 2/ +2/

Округление вниз до числа, кратного 16

m:

03da/0000001111011010/ 986/ +986/

result:

03d0/0000001111010000/ 976/ +976/

n:

ff82/1111111110000010/65410/ -126/

result:

03d0/0000001111010000/ 976/ +976/

**Задание 4**

**int main()**

**{**

**cout<<"\n\nЗадание 4\_\_\_\_\_\_\_\_\_\_"<<endl;**

**signed short m1 = 986;**

**unsigned short m2 = 986;**

**signed short n1 = -126;**

**unsigned short n2 = -126;**

**signed short sd1 = m1 << 1;**

**cout << endl**

**<< "Знаковый сдвиг влево на 1 бит, значение и результат для числа m и "**

**"числа n:"**

**<< endl;**

**cout << "m:" << endl;**

**print16(&m1);**

**cout << "result:" << endl;**

**print16(&sd1);**

**sd1 = n1 << 1;**

**cout << "n:" << endl;**

**print16(&n1);**

**cout << "result:" << endl;**

**print16(&sd1);**

**unsigned short sd2 = m2 << 1;**

**cout << endl**

**<< "Беззнаковый сдвиг влево на 1 бит, значение и результат для числа m и "**

**"числа n:"**

**<< endl;**

**cout << "m:" << endl;**

**print16(&m2);**

**cout << "result:" << endl;**

**print16(&sd2);**

**sd2 = n2 << 1;**

**cout << "n:" << endl;**

**print16(&n2);**

**cout << "result:" << endl;**

**print16(&sd2);**

**signed short sd\_right1 = m1 >> 1;**

**cout << endl**

**<< "Знаковый сдвиг вправо на 1 бит, значение и результат для числа m и "**

**"числа n"**

**<< endl;**

**cout << "m:" << endl;**

**print16(&m1);**

**cout << "result:" << endl;**

**print16(&sd\_right1);**

**sd\_right1 = n1 >> 1;**

**cout << "n:" << endl;**

**print16(&n1);**

**cout << "result:" << endl;**

**print16(&sd\_right1);**

**unsigned short sd\_right2 = m2 >> 1;**

**cout << endl**

**<< "Беззнаковый сдвиг вправо на 1 бит, значение и результат для числа m и "**

**"числа n::"**

**<< endl;**

**cout << "m:" << endl;**

**print16(&m2);**

**cout << "result:" << endl;**

**print16(&sd\_right2);**

**sd\_right2 = n2 >> 1;**

**cout << "n:" << endl;**

**print16(&n2);**

**cout << "result:" << endl;**

**print16(&sd\_right2);**

**unsigned short ost = m2 & 15;**

**cout << endl**

**<< "Рассчёт x & 15 для m и n, значение и результат для числа m "**

**"и числа n:"**

**<< endl;**

**cout << "m:" << endl;**

**print16(&m2);**

**cout << "result:" << endl;**

**print16(&ost);**

**ost = n2 & 15;**

**cout << "n:" << endl;**

**print16(&n2);**

**cout << "result:" << endl;**

**print16(&ost);**

**unsigned int round = m2 & -16;**

**cout << endl << "Рассчёт x & -16 для m и n, значение и результат для числа m "**

**"и числа n:" << endl;**

**cout << "m:" << endl;**

**print16(&m2);**

**cout << "result:" << endl;**

**print16(&round);**

**ost = n2 & -16;**

**cout << "n:" << endl;**

**print16(&n2);**

**cout << "result:" << endl;**

**print16(&round);**

**}**

**Знаковый сдвиг влево на 1 бит, значение и результат для числа m и числа n:**

**m:**

**03da/0000001111011010/ 986/ +986/**

**result:**

**07b4/0000011110110100/ 1972/ +1972/**

**n:**

**ff82/1111111110000010/65410/ -126/**

**result:**

**ff04/1111111100000100/65284/ -252/**

**Беззнаковый сдвиг влево на 1 бит, значение и результат для числа m и числа n:**

**m:**

**03da/0000001111011010/ 986/ +986/**

**result:**

**07b4/0000011110110100/ 1972/ +1972/**

**n:**

**ff82/1111111110000010/65410/ -126/**

**result:**

**ff04/1111111100000100/65284/ -252/**

**Знаковый сдвиг вправо на 1 бит, значение и результат для числа m и числа n**

**m:**

**03da/0000001111011010/ 986/ +986/**

**result:**

**01ed/0000000111101101/ 493/ +493/**

**n:**

**ff82/1111111110000010/65410/ -126/**

**result:**

**ffc1/1111111111000001/65473/ -63/**

**Беззнаковый сдвиг вправо на 1 бит, значение и результат для числа m и числа n::**

**m:**

**03da/0000001111011010/ 986/ +986/**

**result:**

**01ed/0000000111101101/ 493/ +493/**

**n:**

**ff82/1111111110000010/65410/ -126/**

**result:**

**7fc1/0111111111000001/32705/+32705/**

**Рассчёт x & 15 для m и n, значение и результат для числа m и числа n:**

**m:**

**03da/0000001111011010/ 986/ +986/**

**result:**

**000a/0000000000001010/ 10/ +10/**

**n:**

**ff82/1111111110000010/65410/ -126/**

**result:**

**0002/0000000000000010/ 2/ +2/**

**Рассчёт x & -16 для m и n, значение и результат для числа m и числа n:**

**m:**

**03da/0000001111011010/ 986/ +986/**

**result:**

**03d0/0000001111010000/ 976/ +976/**

**n:**

**ff82/1111111110000010/65410/ -126/**

**result:**

**03d0/0000001111010000/ 976/ +976/**

**Задание 5**

**int main() {**

**unsigned int d = 64;**

**unsigned int x = 234;**

**cout << "Округление беззнакового целого числа вниз: " << x << endl;**

**unsigned int remains =**

**(x & (d-1)); // получаем остаток от деления при помощи конъюнкции**

**cout << "Результат: " << x - remains << endl;**

**unsigned int q1 = x + remains ;**

**print16(&q1);**

**cout << "Округление беззнакового целого числа вверх: " << x << endl;**

**remains = ~((x - 1) & (d - 1));**

**remains = (remains & (d - 1));**

**cout << "Результат: " << x + remains << endl;**

**unsigned int q2 = x + remains ;**

**print16(&q2);**

**}**

**Округление беззнакового целого числа вниз: 234**

**Результат: 192**

**0114/0000000100010100/ 276/ +276/**

**Округление беззнакового целого числа вверх: 234**

**Результат: 256**

**0100/0000000100000000/ 256/ +256/**

**Задание 6**

// Type your code here, or load an example.

#include <iostream>

#include <sstream>

#include <bitset>

#include <climits>

#include <cstddef>

#include <iomanip>

**using** **namespace** std;

**void** myIncr(**void** \*p){

(\***reinterpret\_cast**<**unsigned** **int** \*>(p))++;

}

**void** print16(**void** \*p)

{

**unsigned** **short** \*a=**reinterpret\_cast**<**unsigned** **short**\*>(p);

bitset<16> bitform(\*a);

**short** \*b=**reinterpret\_cast**<**short**\*>(p);

cout<<setfill('0');

cout<<setw(4)<<hex<<\*a<<'/'<<setw(16)<<bitform<<'/'<<setfill(' ')<<setw(5)<<dec<<\*a<<'/'<<setfill(' ')<<showpos<<setw(6)

<<\*b<<'/'<<endl<<endl;

}

**void** print32(**void** \*p)

{

**unsigned** **int** \*i = **reinterpret\_cast**<**unsigned** **int** \*>(p);

bitset<32> bitform(\*i);

cout << setfill('0')<< setw(8) << hex << \*i << '/'<< setw(32) << bitform << '/'<< setfill(' ') << setw(10) << dec << \*i << '/';

**int** \*c = **reinterpret\_cast**<**int** \*>(p);

cout << setfill(' ') << showpos << setw(11) << \*c << '/';

**float** \*b = **reinterpret\_cast**<**float** \*>(p);

cout << setfill(' ') << setw(55) << dec << setprecision(50) << fixed << \*b

<< '/';

cout << setfill(' ') << setw(10) << dec << setprecision(3) << scientific

<< \*b << endl

<< endl;

}

**int** main(){

**unsigned** **int** m = 986;

myIncr(&m);

cout << "Инкремент целочисленной беззнаковой переменной m: " << endl;

print32(&m);

**unsigned** **int** n = -126;

myIncr(&n);

cout << "Инкремент целочисленной беззнаковой переменной n: " << endl;

print32(&n);

**signed** **int** m2 = 986;

myIncr(&m2);

cout << "Инкремент целочисленной знаковой переменной m: " << endl;

print32(&m2);

**signed** **int** n2 = -126;

n2++;

cout << "Инкремент целочисленной знаковой переменной n: " << endl;

print32(&n2);

**float** a = 0;

myIncr(&a);

cout << "Инкремент переменной a с плавающей запятой: " << endl;

print32(&a);

**float** b = 1;

b++;

cout << "Инкремент переменной b с плавающей запятой: " << endl;

print32(&b);

**float** c = 12233445;

myIncr(&c);

cout << "Инкремент переменной c с плавающей запятой: " << endl;

print32(&c);

**float** d = 122334455;

d++;

cout << "Инкремент переменной d с плавающей запятой: " << endl;

print32(&d);

**int** count1 = 0;

myIncr(&count1);

cout << "Инкремент нуля: " << endl;

print32(&count1);

**unsigned** **int** count2 = 4294967295;

myIncr(&count2);

cout << "Инкремент максимального целого 32-битного значения без знака: " << endl;

print32(&count2);

**int** count3 = -2147483648;

myIncr(&count3);

cout << "Инкремент минимального целого 32-битного значения со знаком: " << endl;

print32(&count3);

**unsigned** **int** count4 = 2147483647;

myIncr(&count4);

cout << "Инкремент максимального целого 32-битного значения со знаком: " << endl;

print32(&count4);

}

**Инкремент целочисленной беззнаковой переменной m:**

**000003db/00000000000000000000001111011011/ 987/ +987/ +0.00000000000000000000000000000000000000000138308158/+1.383e-42**

**Инкремент цел�численной беззнаковой переменной n:**

**ffffff83/11111111111111111111111110000011/4294967171/ -125/ -nan/ -nan**

**Инкремент целочисленной знаковой переменной m:**

**000003db/00000000000000000000001111011011/ 987/ +987/ +0.00000000000000000000000000000000000000000138308158/+1.383e-42**

**Инкремент целочисленной знаковой переменной n:**

**ffffff83/11111111111111111111111110000011/4294967171/ -125/ -nan/ -nan**

**Инкремент �еременной a с плавающей запятой:**

**00000001/00000000000000000000000000000001/ 1/ +1/ +0.00000000000000000000000000000000000000000000140130/+1.401e-45**

**Инкремент переменной b с плавающей запятой:**

**40000000/01000000000000000000000000000000/1073741824/+1073741824/ +2.00000000000000000000000000000000000000000000000000/+2.000e+00**

**Инкремент переменной c с плавающей запятой:**

**4b3aaae6/01001011001110101010101011100110/1262136038/+1262136038/+12233446.00000000000000000000000000000000000000000000000000/+1.223e+07**

**Инкремент переменн�й d с плавающей запятой:**

**4ce9559f/01001100111010010101010110011111/1290360223/+1290360223/+122334456.00000000000000000000000000000000000000000000000000/+1.223e+08**

**Ин�ремент нуля:**

**00000001/00000000000000000000000000000001/ 1/ +1/ +0.00000000000000000000000000000000000000000000140130/+1.401e-45**

**Инкремент максимального целого 32-битного значения без знака:**

**00000000/00000000000000000000000000000000/ 0/ +0/ +0.00000000000000000000000000000000000000000000000000/+0.000e+00**

**Инкремент минимального целого 32-битного значения со знаком:**

**80000001/10000000000000000000000000000001/2147483649/-2147483647/ -0.00000000000000000000000000000000000000000000140130/-1.401e-45**

**Инкрем�нт максимального целого 32-битного значения со знаком:**

**80000000/10000000000000000000000000000000/2147483648/-2147483648/ -0.00000000000000000000000000000000000000000000000000/-0.000e+00**

**// Type your code here, or load an example.**

**#include <iostream>**

**#include <sstream>**

**#include <bitset>**

**#include <climits>**

**#include <cstddef>**

**#include <iomanip>**

**using namespace std;**

**void myDecr(void \*p){**

**(\*reinterpret\_cast<unsigned int \*>(p))--;**

**}**

**void print16(void \*p)**

**{**

**unsigned short \*a=reinterpret\_cast<unsigned short\*>(p);**

**bitset<16> bitform(\*a);**

**short \*b=reinterpret\_cast<short\*>(p);**

**cout<<setfill('0');**

**cout<<setw(4)<<hex<<\*a<<'/'<<setw(16)<<bitform<<'/'<<setfill(' ')<<setw(5)<<dec<<\*a<<'/'<<setfill(' ')<<showpos<<setw(6)**

**<<\*b<<'/'<<endl<<endl;**

**}**

**void print32(void \*p)**

**{**

**unsigned int \*i = reinterpret\_cast<unsigned int \*>(p);**

**bitset<32> bitform(\*i);**

**cout << setfill('0')<< setw(8) << hex << \*i << '/'<< setw(32) << bitform << '/'<< setfill(' ') << setw(10) << dec << \*i << '/';**

**int \*c = reinterpret\_cast<int \*>(p);**

**cout << setfill(' ') << showpos << setw(11) << \*c << '/';**

**float \*b = reinterpret\_cast<float \*>(p);**

**cout << setfill(' ') << setw(55) << dec << setprecision(50) << fixed << \*b**

**<< '/';**

**cout << setfill(' ') << setw(10) << dec << setprecision(3) << scientific**

**<< \*b << endl**

**<< endl;**

**}**

**int main() {**

**unsigned int m = 986;**

**m--;**

**cout << "Декремент целочисленной беззнаковой переменной m: " << endl;**

**print32(&m);**

**unsigned int n = -126;**

**myDecr(&n);**

**cout << "Декремент целочисленной беззнаковой переменной n: " << endl;**

**print32(&n);**

**signed int m2 = 986;**

**myDecr(&m2);**

**cout << "Декремент целочисленной знаковой переменной m: " << endl;**

**print32(&m2);**

**signed int n2 = -126;**

**myDecr(&n2);**

**cout << "Декремент целочисленной знаковой переменной n: " << endl;**

**print32(&n2);**

**float a = 0;**

**myDecr(&a);**

**cout << "Декремент переменной a с плавающей запятой: " << endl;**

**print32(&a);**

**float b = 1;**

**myDecr(&b);**

**cout << "Декремент переменной b с плавающей запятой: " << endl;**

**print32(&b);**

**float c = 12233445;**

**myDecr(&c);**

**cout << "Декремент переменной c с плавающей запятой: " << endl;**

**print32(&c);**

**float d = 122334455;**

**myDecr(&d);**

**cout << "Декремент переменной d с плавающей запятой: " << endl;**

**print32(&d);**

**int count1 = 0;**

**myDecr(&count1);**

**cout << "Декремент нуля: " << endl;**

**print32(&count1);**

**unsigned int count2 = 4294967295;**

**myDecr(&count2);**

**cout << "Декремент максимального целого 32-битного значения без знака: " << endl;**

**print32(&count2);**

**int count3 = -2147483648;**

**myDecr(&count3);**

**cout << "Декремент минимального целого 32-битного значения со знаком: " << endl;**

**print32(&count3);**

**unsigned int count4 = 2147483647;**

**myDecr(&count4);**

**cout << "Декремент максимального целого 32-битного значения со знаком: " << endl;**

**print32(&count4);**

**}**

**Декремент целочисленной беззнаковой переменной m:**

**000003d9/00000000000000000000001111011001/ 985/ +985/ +0.00000000000000000000000000000000000000000138027899/+1.380e-42**

**Декремент цел�численной беззнаковой переменной n:**

**ffffff81/11111111111111111111111110000001/4294967169/ -127/ -nan/ -nan**

**Декремент целочисленной знаковой переменной m:**

**000003d9/00000000000000000000001111011001/ 985/ +985/ +0.00000000000000000000000000000000000000000138027899/+1.380e-42**

**Декремент целочисленной знаковой переменной n:**

**ffffff81/11111111111111111111111110000001/4294967169/ -127/ -nan/ -nan**

**Декремент �еременной a с плавающей запятой:**

**ffffffff/11111111111111111111111111111111/4294967295/ -1/ -nan/ -nan**

**Декремент переменной b с плавающей запятой:**

**3f7fffff/00111111011111111111111111111111/1065353215/+1065353215/ +0.99999994039535522460937500000000000000000000000000/+1.000e+00**

**Декремент переменной c с плавающей запятой:**

**4b3aaae4/01001011001110101010101011100100/1262136036/+1262136036/+12233444.00000000000000000000000000000000000000000000000000/+1.223e+07**

**Декремент переменн�й d с плавающей запятой:**

**4ce9559e/01001100111010010101010110011110/1290360222/+1290360222/+122334448.00000000000000000000000000000000000000000000000000/+1.223e+08**

**Де�ремент нуля:**

**ffffffff/11111111111111111111111111111111/4294967295/ -1/ -nan/ -nan**

**Декремент максимального целого 32-битного значения без знака:**

**fffffffe/11111111111111111111111111111110/4294967294/ -2/ -nan/ -nan**

**Декремент минимального целого 32-битного значения со знаком:**

**7fffffff/01111111111111111111111111111111/2147483647/+2147483647/ +nan/ +nan**

**Декрем�нт максимального целого 32-битного значения со знаком:**

**7ffffffe/01111111111111111111111111111110/2147483646/+2147483646/ +nan/ +nan**