

федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
«Московский институт электронной техники»

Направление подготовки 09.03.04 «Программная инженерия»
Дисциплина «Электротехника»

Лабораторная РАБОТА №1
Представление данных в ЭВМ

Работу выполнили студенты группы ПИН-24 Баранов Д.А. и Демочкина А.В.
Работу проверил ассистент Института СПИНТех Фомин Р.А.

Цель работы: изучить размеры стандартных типов C/C++ и форматы представления чисел на выбранной платформе.

Вариант:

2	$x = 0x8A8B8C8D, y = 6, z = -3$
---	---------------------------------

Задание №1.

Выберите платформу, на которой планируется в дальнейшем выполнять лабораторные работы.

Для выполнения заданий выбран онлайн компилятор <https://godbolt.org/>.

ОС и разрядность ОС: GNU/Linux 64

Компилятор: Compiler Explorer x86-64 gcc 11.2

Архитектура: x86-64

Задание №2.

При помощи оператора sizeof языка C/C++ выясните, сколько байтов занимают переменные следующих типов языка C/C++: char, signed char, unsigned char, wchar_t, short, unsigned short, int, unsigned int, long, unsigned long, long long, unsigned long long, float, double, long double, size_t, ptrdiff_t, void*, char*, int*, unsigned int* на выбранной платформе.

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char i = '1';
7      cout<<"char - " << sizeof(i)<<" byte "<<endl;
8      cout<<"signed char - "<<sizeof((signed char)i)<<" byte "<<endl;
9      cout<<"unsigned char - "<<sizeof((unsigned char)i)<<" byte "<<endl;
10     cout<<"wchar_t - "<<sizeof((wchar_t)i)<<" byte "<<endl;
11     cout<<"short - "<<sizeof((short)i)<<" byte "<<endl;
12     cout<<"unsigned short - "<<sizeof((unsigned short)i)<<" byte "<<endl;
13     cout<<"int - "<<sizeof((int)i)<<" byte "<<endl;
14     cout<<"unsigned int - "<<sizeof((unsigned int)i)<<" byte "<<endl;
15     cout<<"long - "<<sizeof((long)i)<<" byte "<<endl;
16     cout<<"unsigned long - "<<sizeof((unsigned long)i)<<" byte "<<endl;
17     cout<<"long long - "<<sizeof((long long)i)<<" byte "<<endl;
18     cout<<"unsigned long long - "<<sizeof((unsigned long long)i)<<" byte "<<endl;
19     cout<<"float - "<<sizeof((float)i)<<" byte "<<endl;
20     cout<<"double - "<<sizeof((double)i)<<" byte "<<endl;
21     cout<<"long double - "<<sizeof((long double)i)<<" byte "<<endl;
22     cout<<"size_t - "<<sizeof((size_t)i)<<" byte "<<endl;
23     cout<<"ptrdiff_t - "<<sizeof((ptrdiff_t)i)<<" byte "<<endl;
24     void* p;
25     cout<<"void* - "<<sizeof(p)<<" byte "<<endl;
26     cout<<"char* - "<<sizeof((char*)p)<<" byte "<<endl;
27     cout<<"int* - "<<sizeof((int*)p)<<" byte "<<endl;
28     cout<<"unsigned int* - "<<sizeof((unsigned int*)p)<<" byte "<<endl;
29 }
```

```

char - 1 byte
signed char - 1 byte
unsigned char - 1 byte
wchar_t - 4 byte
short - 2 byte
unsigned short - 2 byte
int - 4 byte
unsigned int - 4 byte
long - 8 byte
unsigned long - 8 byte
long long - 8 byte
unsigned long long - 8 byte
float - 4 byte
double - 8 byte
long double - 16 byte
size_t - 8 byte
ptrdiff_t - 8 byte
void* - 8 byte
char* - 8 byte
int* - 8 byte
unsigned int* - 8 byte

```

```

#include <iostream>
using namespace std;

int main()
{
    char i = '1';
    cout<<"char - " << sizeof(i)<<" byte "<<endl;
    cout<<"signed char - " << sizeof((signed char)i)<<" byte "<<endl;
    cout<<"unsigned char - " << sizeof((unsigned char)i)<<" byte "<<endl;
    cout<<"wchar_t - " << sizeof((wchar_t)i)<<" byte "<<endl;
    cout<<"short - " << sizeof((short)i)<<" byte "<<endl;
    cout<<"unsigned short - " << sizeof((unsigned short)i)<<" byte "<<endl;
    cout<<"int - " << sizeof((int)i)<<" byte "<<endl;
    cout<<"unsigned int - " << sizeof((unsigned int)i)<<" byte "<<endl;
    cout<<"long - " << sizeof((long)i)<<" byte "<<endl;
    cout<<"unsigned long - " << sizeof((unsigned long)i)<<" byte "<<endl;
    cout<<"long long - " << sizeof((long long)i)<<" byte "<<endl;
    cout<<"unsigned long long - " << sizeof((unsigned long long)i)<<" byte "<<endl;
    cout<<"float - " << sizeof((float)i)<<" byte "<<endl;
    cout<<"double - " << sizeof((double)i)<<" byte "<<endl;
    cout<<"long double - " << sizeof((long double)i)<<" byte "<<endl;
    cout<<"size_t - " << sizeof((size_t)i)<<" byte "<<endl;
    cout<<"ptrdiff_t - " << sizeof((ptrdiff_t)i)<<" byte "<<endl;
    void* p;
    cout<<"void* - " << sizeof(p)<<" byte "<<endl;
    cout<<"char* - " << sizeof((char*)p)<<" byte "<<endl;
    cout<<"int* - " << sizeof((int*)p)<<" byte "<<endl;
    cout<<"unsigned int* - " << sizeof((unsigned int*)p)<<" byte "<<endl;
}

```

Целочисленные типы short, unsigned short имеют разрядность 16 бит.

Целочисленные типы int, unsigned int имеют разрядность 32 бита.

Целочисленные типы long, unsigned long, long long, unsigned long long имеют разрядность 64 бита.

Типы с плавающей запятой разрядность 16 бит не имеют.

Тип с плавающей запятой float имеет разрядность 32 бита.

Тип с плавающей запятой double имеет разрядность 64 бита.

Задание №3.

Изучите, как интерпретируется одна и та же область памяти, если она рассматривается как знаковое или беззнаковое целое число, а также — как одно и то же число записывается в различных системах счисления.

Для этого на языке C/C++ разработайте функцию `void print16(void * p)`, которая печатает для 16-битной области памяти по заданному адресу *p*:

- а) целочисленную беззнаковую интерпретацию в шестнадцатеричном представлении;
- б) целочисленную беззнаковую интерпретацию в двоичном представлении;
- в) целочисленную беззнаковую интерпретацию в десятичном представлении;
- г) целочисленную знаковую интерпретацию в шестнадцатеричном представлении;
- д) целочисленную знаковую интерпретацию в двоичном представлении;
- е) целочисленную знаковую интерпретацию в десятичном представлении.

	мин цел без знака =0	макс цел без знака =65535	мин цел со знаком =-32768	макс цел со знаком =32767	y =6	z =-3
а)	0	ffff	8000	7fff	6	fffd
б)	00000000 0000000	11111111 1111111	10000000 0000000	01111111 1111111	00000000 0000110	11111111 1111101
в)	0	65535	32768	32767	6	65533
е)	0	-1	-32768	32767	6	-3

```

#include <iostream>
#include <stdint.h>
#include <typeinfo>
#include <bitset>
using namespace std;

void print16(void *p){
    unsigned short* i = reinterpret_cast<unsigned short *>(p);
    bitset<16> bitform(*i);
    cout<<"а) 16ричная: "<< hex << *i<<endl;
    cout<<"б) 2ичная: " << bitform <<endl;
    cout<<"в) 10ричная: "<< dec<< *i<<endl;
    short* c = reinterpret_cast< short *>(p);
    bitset<16> bitform2(*c);
    cout<<"г) 16ричная: "<< hex << *c<<endl;
    cout<<"д) 2ичная: " << bitform2<<endl;
    cout<<"е) 10ричная: "<< dec<< *c<<endl<<endl;
};

int main(){
    int p;
    p = 0;
    cout<<"мин цел без знака ="<< p <<endl;
    print16(&p);

    p = 65535;
    cout<<"макс цел без знака ="<< p <<endl;
    print16(&p);

    p = -32768;
    cout<<"мин цел со знаком ="<< p <<endl;
    print16(&p);

    p= 32767;
    cout<<"макс цел со знаком ="<< p <<endl;
    print16(&p);

    p=6;
    cout<<"y ="<< p <<endl;
    print16(&p);

    p= -3;
    cout<<"z ="<< p <<endl;
    print16(&p);
}

```

```

мин цел без знака =0
а) 16ричная: 0
б) 2ичная: 0000000000000000
в) 10ричная: 0
г) 16ричная: 0
д) 2ичная: 0000000000000000
е) 10ричная: 0

макс цел без знака =65535
а) 16ричная: ffff
б) 2ичная: 1111111111111111
в) 10ричная: 65535
г) 16ричная: ffff
д) 2ичная: 1111111111111111
е) 10ричная: -1

мин цел со знаком =-32768
а) 16ричная: 8000
б) 2ичная: 1000000000000000
в) 10ричная: 32768
г) 16ричная: 8000
д) 2ичная: 1000000000000000
е) 10ричная: -32768

макс цел со знаком =32767
а) 16ричная: 7fff
б) 2ичная: 0111111111111111
в) 10ричная: 32767
г) 16ричная: 7fff
д) 2ичная: 0111111111111111
е) 10ричная: 32767

y =6
а) 16ричная: 6
б) 2ичная: 0000000000000110
в) 10ричная: 6
г) 16ричная: 6
д) 2ичная: 0000000000000110
е) 10ричная: 6

z =-3
а) 16ричная: fffd
б) 2ичная: 111111111111101
в) 10ричная: 65533
г) 16ричная: fffd
д) 2ичная: 111111111111101
е) 10ричная: -3

#include <iostream>
#include <stdint.h>
#include <typeinfo>
#include <bitset>
using namespace std;

void print16(void *p){
    unsigned short* i = reinterpret_cast<unsigned short *>(p);
    bitset<16> bitform(*i);
    cout<<"а) 16ричная: "<< hex << *i<<endl;
    cout<<"б) 2ичная: " << bitform <<endl;
    cout<<"в) 10ричная: "<< dec<< *i<<endl;
    short* c = reinterpret_cast< short *>(p);
    bitset<16> bitform2(*c);
    cout<<"г) 16ричная: "<< hex << *c<<endl;

```

```

        cout<<"д) 2ичная: " << bitform2<<endl;
        cout<<"е) 10ричная: "<< dec<< *c<<endl<<endl;

};

int main() {
    int p;
    p = 0;
    cout<<"мин цел без знака ="<< p <<endl;
    print16(&p);

    p = 65535;
    cout<<"макс цел без знака ="<< p <<endl;
    print16(&p);

    p = -32768;
    cout<<"мин цел со знаком ="<< p <<endl;
    print16(&p);

    p= 32767;
    cout<<"макс цел со знаком ="<< p <<endl;
    print16(&p);

    p=6;
    cout<<"y ="<< p <<endl;
    print16(&p);

    p= -3;
    cout<<"z ="<< p <<endl;
    print16(&p);
}

```

В результате выполнения программы мы убедились, что шестнадцатеричное представление одно и тоже. Также двоичное представление одно и тоже.

Задание №4.

Разработайте на языке C/C++ функции *print32()* и *print64()*, аналогичные *print16()* для размеров 32 и 64 бита, и дополните их интерпретацией памяти как числа с плавающей запятой соответствующего размера. Необходимо напечатать:

- ж) интерпретацию с плавающей запятой в представлении с фиксированным количеством цифр после запятой;
- з) интерпретацию с плавающей запятой в экспоненциальном представлении.

Функция *print32()*:

	мин цел без знака a =0	макс цел без знака a =429 4967 295	мин цел со знаком =-2147 483648	макс цел со знаком =21474 83647	x = 0x8a8b 8c8d	y =6	z =-3	float x = 0x8a8b 8c8d	float y =6	float z =-3
a)	0	ffff ffff	800000 00	7ffffff ff	8a8b8c 8d	6	ffffff fd	8a8b8c 8d	40c000 00	c04000 00

б)	0000 0000 0000 0000 0000 0000 0000 0000 0000	1111 1111 1111 1111 1111 1111 1111 1111 1111	100000 000000 000000 000000 000000 000000 00 00	011111 111111 111111 111111 111111 111111 11 11	100010 101000 101110 001100 100011 01	0000 0000 0000 0000 0000 0000 0000 0000 0110	111111 111111 111111 111111 111111 111111 01 01	100010 101000 101110 001100 100011 01	010000 001100 000000 000000 000000 000000 00 00	110000 000100 000000 000000 000000 000000 00 00
в)	0	4294 9672 95	214748 3648	214748 3647	232440 1293	6	429496 7293	232440 1293	108632 4736	322541 9776
е)	0	-1	-21474 83648	214748 3647	-19705 66003	6	-3	-19705 66003	108632 4736	-10695 47520
ж)	0.00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000	-nan	-0.000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000	nan	-0.000 000000 000000 000000 000000 000000 000013 438082 441287 46009	0.00 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0084 0779	-nan	-0.000 000000 000000 000000 000000 000000 000000 000013 438082 441287 46009	6.0000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000	-3.000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000
з)	0.00 0e+0 0	-nan	-0.000 e+00	nan	-1.344 e-32	8.40 8e-4 5	-nan	-1.344 e-32	6.000e +00	-3.000 e+00


```

#include <iostream>
#include <stdint.h>
#include <typeinfo>
#include <bitset>
#include <iomanip>
using namespace std;

void print32(void *p){
    unsigned int* i = reinterpret_cast<unsigned int *>(p);
    bitset<32> bitform(*i);
    cout<<"a) 16ричная: "<< hex << *i<<endl;
    cout<<"б) 2ичная: " << bitform <<endl;
    cout<<"в) 10ричная: "<< dec<< *i<<endl;
    int* c = reinterpret_cast< int *>(p);
    bitset<32> bitform2(*c);
    cout<<"г) 16ричная: "<< hex << *c<<endl;
    cout<<"д) 2ичная: " << bitform2<<endl;
    cout<<"е) 10ричная: "<< dec<< *c<<endl;
    float* b = reinterpret_cast< float *>(p);
    cout<<"ж) с плавающей запятой с фиксированным количеством цифр: " << dec << setprecision(50) << fixed << *b << endl;
    cout<<"з) с плавающей запятой в экспоненциальном представлении: " << dec << setprecision(3) << scientific << *b << endl << endl;
};

```

```

int main(){
    long p;
    p = 0;
    cout<<"мин цел без знака ="<< p <<endl;
    print32(&p);

    p = 4294967295;
    cout<<"макс цел без знака ="<< p <<endl;
    print32(&p);

    p = -2147483648;
    cout<<"мин цел со знаком ="<< p <<endl;
    print32(&p);

    p= 2147483647;
    cout<<"макс цел со знаком ="<< p <<endl;
    print32(&p);
}

```

```

int *q = (int*)0x8A8B8C8D;
cout<<"1. x ="<< dec << q << endl;
print32(&q);

```

```

p=6;
cout<<"1. y ="<< p <<endl;
print32(&p);

```

```

p= -3;
cout<<"1. z ="<< p <<endl;
print32(&p);

```

```

float *q2 = (float*)0x8A8B8C8D;
cout<<"2. x =" << q2 << endl;
print32(&q2);

```

```

float p2;
p2 = 6.0;
cout<<"2. y ="<< p2 <<endl;
print32(&p2);

```

```

p2 = -3.0;
cout<<"2. float z ="<< p2 <<endl;
print32(&p2);

```

```

#include <iostream>
#include <stdint.h>
#include <typeinfo>

```

```

#include <bitset>
#include <iomanip>
using namespace std;

void print32(void *p){
    unsigned int* i = reinterpret_cast<unsigned int *>(p);
    bitset<32> bitform(*i);
    cout<<"а) 16ричная: "<< hex << *i<<endl;
    cout<<"б) 2ичная: " << bitform <<endl;
    cout<<"в) 10ричная: "<< dec<< *i<<endl;
    int* c = reinterpret_cast< int *>(p);
    bitset<32> bitform2(*c);
    cout<<"г) 16ричная: "<< hex << *c<<endl;
    cout<<"д) 2ичная: " << bitform2<<endl;
    cout<<"е) 10ричная: "<< dec<< *c<<endl;
    float* b = reinterpret_cast< float *>(p);
    cout<<"ж) с плавающей запятой с фиксированным количеством цифр: " << dec << setprecision(50) <<
fixed << *b << endl;
    cout<<"з) с плавающей запятой в экспоненциальном представлении: " << dec << setprecision(3) <<
scientific << *b << endl << endl;
};

int main(){
    long p;
    p = 0;
    cout<<"мин цел без знака ="<< p <<endl;
    print32(&p);

    p = 4294967295;
    cout<<"макс цел без знака ="<< p <<endl;
    print32(&p);

    p = -2147483648;
    cout<<"мин цел со знаком ="<< p <<endl;
    print32(&p);

    p= 2147483647;
    cout<<"макс цел со знаком ="<< p <<endl;
    print32(&p);

    int *q = (int*)0x8A8B8C8D;
    cout<<"1. x ="<< dec << q << endl;
    print32(&q);

    p=6;
    cout<<"1. y ="<< p <<endl;
    print32(&p);

    p= -3;
    cout<<"1. z ="<< p <<endl;
    print32(&p);

    float *q2 = (float*)0x8A8B8C8D;
    cout<<"2. x =" << q2 << endl;
    print32(&q2);

    float p2;
    p2 = 6.0;

```


[illegible]

Функция print64():

	мин цел без знака =0	макс цел без знак а =184 4674 4073 7095 5161 5	мин цел со знаком =-9223 372036 854775 808	макс цел со знаком =92233 720368 547758 07	x = 0x8a8b 8c8d	y =6	z =-3	double x = 0x8a8b 8c8d	double y =6	double z =-3
а)	0	ffff ffff ffff ffff	800000 000000 0000	7ffffff ffffff ffff	8a8b8c 8d	6	ffffff ffffff ffffff d	8a8b8c 8d	8a8b8c 8d40c0 0000	c00800 000000 0000
б)	00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 0000	1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111	100000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 0000	011111 111111 111111 111111 111111 111111 111111 111111 111111 111111 111111 111111 111111 111111 111111 111111 111111 1111	000000 000000 000000 000000 000000 000000 000000 000000 001000 101010 001011 100011 001000 1101	00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 0110	11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 11111 1101	000000 000000 000000 000000 000000 000000 000000 000000 001000 101010 001011 100011 001000 1101	010000 000001 100000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 0000	110000 000000 100000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 0000
в)	0	1844 6744 0737 0955 1615	922337 203685 477580 8	922337 203685 477580 7	232440 1293	6	18446 74407 37095 51613	232440 1293	461844 141786 844364 8	138373 098550 958489 60
е)	0	-1	-92233 720368 547758 08	922337 203685 477580 7	232440 1293	6	-3	232440 1293	461844 141786 844364 8	-46094 342186 137026 56
ж)	0.000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00	-nan	-0.000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 00000	nan	0.0000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 0000	0.000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00	-nan	0.0000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 0000	6.0000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 00000 0000	-3.000 000000 000000 000000 000000 000000 000000 000000 000000 000000 000000 00000 00000
з)	0.000 e+00	-nan	-0.000 e+00	nan	1.148e -314	2.964 e-323	-nan	1.148e -314	6.000e +00	-3.000 e+00

```

#include <iostream>
#include <stdint.h>
#include <typeinfo>
#include <bitset>
#include <iomanip>
using namespace std;

void print64(void *p){
    unsigned long* i = reinterpret_cast<unsigned long*>(p);
    bitset<64> bitform(*i);
    cout<<"a) 16ричная: "<< hex << *i<<endl;
    cout<<"б) 2ичная: " << bitform <<endl;
    cout<<"в) 10ричная: "<< dec<< *i<<endl;
    long* c = reinterpret_cast< long*>(p);
    bitset<64> bitform2(*c);
    cout<<"г) 16ричная: "<< hex << *c<<endl;
    cout<<"д) 2ичная: " << bitform2<<endl;
    cout<<"е) 10ричная: "<< dec<< *c<<endl;
    double* b = reinterpret_cast< double*>(p);
    cout<<"ж) с плавающей запятой с фиксированным количеством цифр: " << dec << setprecision(50) << fixed << *b << endl;
    cout<<"з) с плавающей запятой в экспоненциальном представлении: " << dec << setprecision(3) << scientific << *b << endl << endl;
};

int main(){
    unsigned long p;
    p = 0;
    cout<<"мин цел без знака ="<< p <<endl;
    print64(&p);

    p = 18446744073709551615;
    cout<<"макс цел без знака ="<< p <<endl;
    print64(&p);

    long p2;
    p2 = -9223372036854775808;
    cout<<"мин цел со знаком ="<< p2 <<endl;
    print64(&p2);

    p2 = 9223372036854775807;
    cout<<"макс цел со знаком ="<< p2 <<endl;
    print64(&p2);

    long *q = (long*)0x8A8B8C8D;
    cout<<"1. x ="<< dec << q << endl;
    print64(&q);

    p2 = 6;
    cout<<"1. y ="<< p2 <<endl;
    print64(&p2);

    p2 = -3;
    cout<<"1. z ="<< p2 <<endl;
    print64(&p2);

    double *q2 = (double*)0x8A8B8C8D;
    cout<<"2. x =" << q2 << endl;
    print64(&q2);

    double p3;
    p3 = 6.0;
    cout<<"2. y ="<< p3 <<endl;
    print64(&p3);

    p3 = -3.0;
    cout<<"2. float z ="<< p3 <<endl;
    print64(&p3);

#include <iostream>
#include <stdint.h>
#include <typeinfo>
#include <bitset>

```

```

#include <iomanip>
using namespace std;

void print64(void *p){
    unsigned long* i = reinterpret_cast<unsigned long *>(p);
    bitset<64> bitform(*i);
    cout<<"а) 16ричная: "<< hex << *i<<endl;
    cout<<"б) 2ичная: " << bitform <<endl;
    cout<<"в) 10ричная: "<< dec<< *i<<endl;
    long* c = reinterpret_cast< long *>(p);
    bitset<64> bitform2(*c);
    cout<<"г) 16ричная: "<< hex << *c<<endl;
    cout<<"д) 2ичная: " << bitform2<<endl;
    cout<<"е) 10ричная: "<< dec<< *c<<endl;
    double* b = reinterpret_cast< double *>(p);
    cout<<"ж) с плавающей запятой с фиксированным количеством цифр: " << dec << setprecision(50) <<
fixed << *b << endl;
    cout<<"з) с плавающей запятой в экспоненциальном представлении: " << dec << setprecision(3) <<
scientific << *b << endl << endl;
};

int main(){
    unsigned long p;
    p = 0;
    cout<<"мин цел без знака ="<< p <<endl;
    print64(&p);

    p = 18446744073709551615;
    cout<<"макс цел без знака ="<< p <<endl;
    print64(&p);

    long p2;
    p2 = -9223372036854775808;
    cout<<"мин цел со знаком ="<< p2 <<endl;
    print64(&p2);

    p2 = 9223372036854775807;
    cout<<"макс цел со знаком ="<< p2 <<endl;
    print64(&p2);

    long *q = (long*)0x8A8B8C8D;
    cout<<"1. x ="<< dec << q << endl;
    print64(&q);

    p2 = 6;
    cout<<"1. y ="<< p2 <<endl;
    print64(&p2);

    p2 = -3;
    cout<<"1. z ="<< p2 <<endl;
    print64(&p2);

    double *q2 = (double*)0x8A8B8C8D;
    cout<<"2. x =" << q2 << endl;
    print64(&q2);

    double p3;
    p3 = 6.0;

```


0.00

1. $y = 6$

[illegible]

г) 16ричная: 6

е) 10ричная: 6

0.000

1. $z = -3$

[illegible]

г) 16ричная: ffffffffffffffff

е) 10ричная: -3

з) с плавающей запятой в экспоненциальном представлении: -nan

а) 16ричная: 8a8b8c8d

в) 10ричная: 2324401293

[illegible]

ж) с плавающей запятой с фиксированным количеством цифр:

з) с плавающей запятой в экспоненциальном представлении: 1.148e-314

а) 16ричная: 4018000000000000

в) 10ричная: 4618441417868443648

д) 2ичная: 0100000000011000

ж) с плавающей запятой с фиксированным количеством цифр:

з) с плавающей запятой в экспоненциальном представлении: 6.000e+00

а) 16ричная: c008000000000000

б) 2ичная: 1100000000001000

в) 10ричная: 13837309855095848960

г) 16ричная: c008000000000000

д) 2ичная: 1100000000001000

е) 10ричная: -4609434218613702656

ж) с плавающей запятой с фиксированным количеством цифр:

[illegible]

з) с плавающей запятой в экспоненциальном представлении: $-3.000e+00$

Задание №5.

Изучите, как располагаются в памяти байты, составляющие целое число и число с плавающей запятой. Для этого на языке C/C++ разработайте функцию `void printDump(void *p, size_t N)`, которая печатает для области памяти по заданному адресу `p` значения `N` байтов, начиная с младшего, в шестнадцатеричном представлении (шестнадцатеричный дамп памяти).

```
#include <iostream>
#include <stdint.h>
#include <typeinfo>
#include <bitset>
#include <iomanip>
using namespace std;

void printDump(void *p, size_t N){
    cout << hex << setfill('0');
    for (size_t i = 0; i < N; i++)
        cout << ' ' << setw(2) << (int)reinterpret_cast<unsigned char *>(p)[i];
    cout << endl;
};
```

```
int main(){
    int x = 0x8A8B8C8D;
    printDump(&x, sizeof x);
    double y = 6;
    printDump(&y, sizeof y);
    long mas[] = {0x8A8B8C8D, 6, -3};
    printDump(&mas, sizeof mas);
}
```

```
#include <iostream>
#include <stdint.h>
#include <typeinfo>
#include <bitset>
#include <iomanip>
using namespace std;

void printDump(void *p, size_t N){
    cout << hex << setfill('0');
    for (size_t i = 0; i < N; i++)
        cout << ' ' << setw(2) << (int)reinterpret_cast<unsigned char *>(p)[i];
    cout << endl;
};

int main(){
    int x = 0x8A8B8C8D;
    printDump(&x, sizeof x);
    double y = 6;
    printDump(&y, sizeof y);
    long mas[] = {0x8A8B8C8D, 6, -3};
    printDump(&mas, sizeof mas);
}
```

```

8d 8c 8b 8a
00 00 00 00 00 00 18 40
8d 8c 8b 8a 00 00 00 00 06 00 00 00 00 00 00 00 fd ff ff ff ff ff ff ff

```

Из результатов выполнения функции видно, что порядок следования байтов в словах для нашего процессора прямой (младший байт по младшему адресу).

Вопросы

1. Как представляются целые числа со знаком и без знака?

Беззнаковые целые числа представляются в виде двоичного представления. Либо в виде восьмеричной и шестнадцатеричной для более компактной записи двоичного кода.

Для чисел со знаком под шестнадцатеричным представлением числа подразумевается представление двоичного содержимого памяти, а не самих данных. Н-р, $-1 = \text{FFFF FFFF}$ обозначает, что значение -1 представляется в памяти двоичным кодом $1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111$, что в шестнадцатеричном виде соответствует FFFF FFFF .

Знаковые целые числа представляются в виде: величины со знаком, кода с избытком (какой-то избыток), дополнительного кода.

Величина со знаком: выделить один бит для хранения знака, а в оставшихся хранить абсолютную величину (модуль) числа. Соответственно, код в виде величины со знаком не используется для целых чисел. Идея раздельного кодирования знака и абсолютной величины используется при кодировании вещественных чисел с плавающей запятой.

Код с избытком: можно задать некоторую константу w и поставить в соответствие каждому знаковому числу x беззнаковое значение $u = x + w$. Код с избытком используется для представления порядка вещественных чисел с плавающей запятой, а также в специальной аппаратуре или для передачи данных по каналам связи, если диапазон данных невелик, но заведомо несимметричен относительно нуля (в частности, год или температура в помещении в градусах Цельсия)

2. Как перевести число в дополнительный код?

Необходимо записывать отрицательное число $-x$ тем кодом, который получается в результате беззнакового вычитания $0 - x$ (с учётом цикличности сложения и вычитания в ЭВМ беззнаковое представление этого кода $2^N - x$). Дополнительный код неотрицательных чисел совпадает с прямым (натуральным).

Дополнительный код $-x$ может быть рассчитан как $\neg x + 1$, где $\neg x$ — побитовое отрицание (инверсия битов) натурального двоичного представления абсолютной величины числа x .

Код позволяет заменить операцию вычитания на операцию сложения и сделать операции сложения и вычитания одинаковыми для знаковых и беззнаковых чисел.

Именно в дополнительном коде представлены отрицательные числа в современных вычислительных системах.

3. Для чего нужно знать порядок следования байтов на вашем компьютере?

Знание порядка байтов необходимо при передаче сведений, которые должны интерпретироваться сетью. Необходимо преобразовывать порядок данных, если компьютеры на обоих концах имеют разный порядок следования байтов, чтобы данные на обоих устройствах считывались одинаково.

Обратный порядок байтов позволяет легко читать числа, записанные в памяти. Обратный порядок принят в протоколе TCP/IP. Прямой порядок байтов удобен при обработке чисел большой разрядности с помощью процессора малой разрядности, так как позволяет при сложении таких чисел обращаться к памяти последовательно в порядке возрастания адресов, а такие запросы выполняются быстрее.

4. Как располагаются в памяти элементы массива?

Последовательно друг за другом, сначала представляется первый элемент массива, затем представляется второй и так далее.

5. Как найти размер массива, зная размер элемента и их количество?

Перемножить размер элемента с их количеством.