

**Министерство науки и образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
"Московский институт электронной техники"
(МИЭТ)**

Отчет по лабораторной работе № 1

"Представление данных в ЭВМ"

Выполнили: студенты ПМ - 31

Алтухова Анна Валерьевна

Мартынова Мария Олеговна

Задание 1.1

Задание:

При помощи оператора sizeof языка C/C++ выясните, сколько байтов занимают переменные следующих типов языка C/C++:

char, signed char, unsigned char, char*, wchar_t, wchar_t*, short, short*, int, int*, long, long*, long long, long long*, float, float*, double, double*, long double, long double* на платформах, доступных на ВЦ (штраф –2 балла за платформу, если в аудитории она доступна, а данных по ней нет; если платформа недоступна в лабораторной аудитории либо её убрали с ВЦ вообще, штраф не начисляется):

ОС	Компилятор	Разрядность сборки
GNU/Linux 64	GCC	64
MS Windows 64	GCC (MinGW)	64
MS Windows 64	GCC (MinGW)	32
MS Windows 64	Microsoft	64
MS Windows 64	Microsoft	32

Результаты оформите в отчёте в виде таблицы

Листинг:

```
template<typename T> //шаблонная функция для печати информации о размере определенного типа данных
void print_size(string type_name)
{
    cout << type_name << ": " << (sizeof(T)) << endl;
}

int main()
{
    cout<<"-----Задание 1-----"<<endl;
    cout<<"Сведения о размере памяти, занимаемой различными типами данных:"<<endl;
    print_size<char>("char");
    print_size<signed char>("signed char");
    print_size<unsigned char>("unsigned char");
    print_size<char*>("char*");
    print_size<wchar_t>("wchar_t");
    print_size<wchar_t*>("wchar_t*");
    print_size<short>("short");
    print_size<short*>("short*");
    print_size<int>("int");
    print_size<int*>("int*");
    print_size<long>("long");
    print_size<long*>("long*");
    print_size<long long>("long long");
    print_size<long long*>("long long*");
    print_size<float>("float");
    print_size<float*>("float*");
    print_size<double>("double");
    print_size<double*>("double*");
    print_size<long double>("long double");
    print_size<long double*>("long double*");
    cout<<"-----"<<endl;
```

Пример вывода (для clang++-7, Linux x64 (Repl.it)):

lab1.1:

-----Задание 1-----

Сведения о размере памяти, занимаемой различными типами данных:

char: 1
signed char: 1
unsigned char: 1
char*: 8
wchar_t: 4
wchar_t*: 8
short: 2
short*: 8
int: 4
int*: 8
long: 8
long*: 8
long long: 8
long long*: 8
float: 4
float*: 8
double: 8
double*: 8
long double: 16
long double*: 8

Тип данных	Размер (Ubuntu x64, clang++-7), байт
char	1
signed char	1
unsigned char	1
char*	8
wchar_t	4
wchar_t*	8
short	2
short*	8
int	4
int*	8
long	8
long*	8
long long	8
long long*	8
float	4
float*	8
double	8
double*	8
long double	16
long double*	8

Задание 1.2

Задание:

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 (и опционально двоичном) представлении;
- б) целочисленную беззнаковую интерпретацию в десятичном представлении;
- в) целочисленную знаковую интерпретацию в шестнадцатеричном (и опционально двоичном) представлении;
- г) целочисленную знаковую интерпретацию в десятичном представлении.

Проверьте работу функции `print16()` на 16-битных целочисленных переменных,

принимающих следующие значения:

- минимальное целое 16-битное значение без знака;
- максимальное целое 16-битное значение без знака;
- минимальное целое 16-битное значение со знаком;
- максимальное целое 16-битное значение со знаком;
- значение *y*, соответствующее варианту (таблица Л1.1);
- значение *z*, соответствующее варианту (таблица Л1.1);

(запишите каждое из значений в 16-битную целочисленную переменную и передайте

её адрес функции).

Убедитесь, что (а) и (в) — одно и то же представление.

Полученные результаты внесите в отчёт в таблицу, каждая строка которой соответствует значению, столбец — представлениям (а), (б), (г).

Листинг:

```
template<class T>
void __print(string type, T value)
{
    cout << type << "16-ичное представление: " << hex << value << endl;
    cout << type << "10-ичное представлении: " << dec << value << endl;
}

void print16(void *p)
{
    __print("Целочисленная беззнаковая интерпретация, ", *(reinterpret_cast<unsigned short*>(p)));
    __print("Целочисленная знаковая интерпретация, ", *(reinterpret_cast<short*>(p)));
    cout<<endl;
}

int main()
{
    setlocale(0, "");
    cout<<"-----Задание 2-----"<<endl;
    unsigned int min_unsigned_16bit=0b0000000000000000;
    unsigned int max_unsigned_16bit=0b1111111111111111;
    int min_signed_16bit=0b1000000000000000;
    int max_signed_16bit=0b0111111111111111;
    int y=3;
    int z=-8;
    cout<<"Представления числа 0b0000000000000000"<<endl;
    print16(&min_unsigned_16bit);
    cout<<"Представления числа 0b1111111111111111"<<endl;
    print16(&max_unsigned_16bit);
    cout<<"Представления числа 0b1000000000000000"<<endl;
    print16(&min_signed_16bit);
    cout<<"Представления числа 0b0111111111111111"<<endl;
    print16(&max_signed_16bit);
    cout<<"Представления числа "<<y<<endl;
    print16(&y);
    cout<<"Представления числа "<<z<<endl;
    print16(&z);
    cout<<"-----"<<endl;

    cout<<"-----Задание 3-----"<<endl;
}
```

Пример вывода:

-----Задание 2-----

Представления числа 0b0000000000000000

Целочисленная беззнаковая интерпретация,16-ичное представление: 0

Целочисленная беззнаковая интерпретация,10-ичное представлении: 0

Целочисленная знаковая интерпретация,16-ичное представление: 0

Целочисленная знаковая интерпретация,10-ичное представлении: 0

Представления числа 0b1111111111111111

Целочисленная беззнаковая интерпретация,16-ичное представление: ffff

Целочисленная беззнаковая интерпретация,10-ичное представлении: 65535

Целочисленная знаковая интерпретация,16-ичное представление: ffff
Целочисленная знаковая интерпретация,10-ичное представлении: -1

Представления числа 0b1000000000000000

Целочисленная беззнаковая интерпретация,16-ичное представление: 8000
Целочисленная беззнаковая интерпретация,10-ичное представлении: 32768
Целочисленная знаковая интерпретация,16-ичное представление: 8000
Целочисленная знаковая интерпретация,10-ичное представлении: -32768

Представления числа 0b0111111111111111

Целочисленная беззнаковая интерпретация,16-ичное представление: 7fff
Целочисленная беззнаковая интерпретация,10-ичное представлении: 32767
Целочисленная знаковая интерпретация,16-ичное представление: 7fff
Целочисленная знаковая интерпретация,10-ичное представлении: 32767

Представления числа 3

Целочисленная беззнаковая интерпретация,16-ичное представление: 3
Целочисленная беззнаковая интерпретация,10-ичное представлении: 3
Целочисленная знаковая интерпретация,16-ичное представление: 3
Целочисленная знаковая интерпретация,10-ичное представлении: 3

Представления числа -8

Целочисленная беззнаковая интерпретация,16-ичное представление: fff8
Целочисленная беззнаковая интерпретация,10-ичное представлении: 65528
Целочисленная знаковая интерпретация,16-ичное представление: fff8
Целочисленная знаковая интерпретация,10-ичное представлении: -8

bin	unsigned hex	unsigned dec	signed hex	signed dec
0000000000000000	0	0	0	0
1111111111111111	ffff	65535	ffff	-1
1000000000000000	8000	32768	8000	-32768
0111111111111111	7fff	32767	7fff	32767
0000000000000011	3	3	3	3
1111111111111000	fff8	65528	fff8	-8

Задание 1.3

Задание:

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:

– *print32()* печатает для 32-битной области памяти по заданному адресу *p* целочисленные представления (а)–(г) и 32-битное представление с плавающей запятой (с одинарной точностью) д).

– *print64()* печатает для 64-битной области памяти по заданному адресу *p* целочисленные представления (а)–(г) и 64-битное представление с плавающей запятой (с двойной точностью) д).

Проверьте работу функций на целочисленных переменных соответствующего размера, принимающих значения:

- минимальное целое значение без знака соответствующего размера;
- максимальное целое значение без знака соответствующего размера;
- минимальное целое значение со знаком соответствующего размера;
- максимальное целое значение со знаком соответствующего размера;
- значение *x*, соответствующее варианту;
- значение *y*, соответствующее варианту;
- значение *z*, соответствующее варианту;

и переменных с плавающей запятой соответствующего размера, принимающих значения:

- значение *x*, соответствующее варианту;
- значение *y*, соответствующее варианту;
- значение *z*, соответствующее варианту;

(*x*, *y*, *z* смотрите в таблице Л1.1).

Выпишите в отчёт полученные результаты (дополните таблицу задания Л1.№2

столбцом д).

Листинг:

```
template<class T>
void __print(string type, T value)
{
    cout << type << "16-ичное представление: " << hex << value << endl;
    cout << type << "10-ичное представлении: " << dec << value << endl;
}
```



```

void print32(void* p)
{
    __print("Целочисленная беззнаковая интерпретация, ", *(reinterpret_cast<unsigned int*>(p)));
    __print("Целочисленная знаковая интерпретация, ", *(reinterpret_cast<int*>(p)));
    cout<<"Интерпретация в качестве числа с плавающей запятой:"<<*(reinterpret_cast<float*>(p))<<endl<<endl;
}

void print64(void* p)
{
    __print("Целочисленная беззнаковая интерпретация, ", *(reinterpret_cast<unsigned long long*>(p)));
    __print("Целочисленная знаковая интерпретация, ", *(reinterpret_cast<long long*>(p)));
    cout<<"Интерпретация в качестве числа с плавающей запятой:"<<*(reinterpret_cast<double*>(p))<<endl<<endl;
}

int main()
{
    setlocale(0, "");
    cout<<"-----Задание 3-----"<<endl;
    //32
    unsigned int min_unsigned_32bit=0b00000000000000000000000000000000;
    unsigned int max_unsigned_32bit=0b11111111111111111111111111111111;
    int min_signed_32bit=0b10000000000000000000000000000000;
    int max_signed_32bit=0b01111111111111111111111111111111;
    cout<<"Представления числа 0b00000000000000000000000000000000"<<endl;
    print32(&min_unsigned_32bit);
    cout<<"Представления числа 0b11111111111111111111111111111111"<<endl;
    print32(&max_unsigned_32bit);
    cout<<"Представления числа 0b10000000000000000000000000000000"<<endl;
    print32(&min_signed_32bit);
    cout<<"Представления числа 0b01111111111111111111111111111111"<<endl;
    print32(&max_signed_32bit);

    int ix = 0xA1B2C3D4;
    int iy = 3;
    int iz = -8;
    cout <<"Представление числа 0xA1B2C3D4 как целочисленного:" <<endl;
    print32(&ix);
    cout <<"Представление числа 3 как целочисленного:"<< endl;
    print32(&iy);
    cout <<"Представление числа -8 как целочисленного:"<< endl;
    print32(&iz);
    cout << endl;

    float fx = 0xA1B2C3D4;
    float fy = 3;
    float fz = -8;
    cout <<"Представление числа 0xA1B2C3D4 с плавающей запятой:" <<endl;
    print32(&fx);
    cout <<"Представление числа 3 с плавающей запятой:"<< endl;
    print32(&fy);
    cout <<"Представление числа -8 с плавающей запятой:" <<endl;
    print32(&fz);
    cout << endl;

    // 64
    unsigned long long min_unsigned_64bit=0b0000000000000000000000000000000000000000000000000000000000000000;
    unsigned long long max_unsigned_64bit=0b1111111111111111111111111111111111111111111111111111111111111111;
    long long min_signed_64bit=0b1000000000000000000000000000000000000000000000000000000000000000;
    long long max_signed_64bit=0b0111111111111111111111111111111111111111111111111111111111111111;
    cout<<"Представления числа 0b0000000000000000000000000000000000000000000000000000000000000000"<<endl;
    print64(&min_unsigned_64bit);
    cout<<"Представления числа 0b1111111111111111111111111111111111111111111111111111111111111111"<<endl;
    print64(&max_unsigned_64bit);
    cout<<"Представления числа 0b1000000000000000000000000000000000000000000000000000000000000000"<<endl;
    print64(&min_signed_64bit);
    cout<<"Представления числа 0b0111111111111111111111111111111111111111111111111111111111111111"<<endl;
    print64(&max_signed_64bit);
}

```



```

long long lx = 0xA1B2C3D4;
long long ly = 3;
long long lz = -8;
cout << "Представление числа 0xA1B2C3D4 как целочисленного:" << endl;
print64(&lx);
cout << "Представление числа 3 как целочисленного:" << endl;
print64(&ly);
cout << "Представление числа -8 как целочисленного:" << endl;
print64(&lz);
cout << endl;

double dx = 0xA1B2C3D4;
double dy = 3;
double dz = -8;
cout << "Представление числа 0xA1B2C3D4 с плавающей запятой:" << endl;
print64(&dx);
cout << "Представление числа 3 с плавающей запятой:" << endl;
print64(&dy);
cout << "Представление числа -8 с плавающей запятой:" << endl;
print64(&dz);
cout << endl;
}

```

Пример вывода:

-----Задание 3-----

Представления числа 0b00000000000000000000000000000000

Целочисленная беззнаковая интерпретация,16-ичное представление: 0

Целочисленная беззнаковая интерпретация,10-ичное представлении: 0

Целочисленная знаковая интерпретация,16-ичное представление: 0

Целочисленная знаковая интерпретация,10-ичное представлении: 0

Интерпретация в качестве числа с плавающей запятой:0

Представления числа 0b11111111111111111111111111111111

Целочисленная беззнаковая интерпретация,16-ичное представление: ffffffff

Целочисленная беззнаковая интерпретация,10-ичное представлении: 4294967295

Целочисленная знаковая интерпретация,16-ичное представление: ffffffff

Целочисленная знаковая интерпретация,10-ичное представлении: -1

Интерпретация в качестве числа с плавающей запятой:nan

Представления числа 0b10000000000000000000000000000000

Целочисленная беззнаковая интерпретация,16-ичное представление: 80000000

Целочисленная беззнаковая интерпретация,10-ичное представлении: 2147483648

Целочисленная знаковая интерпретация,16-ичное представление: 80000000

Целочисленная знаковая интерпретация,10-ичное представлении: -2147483648

Интерпретация в качестве числа с плавающей запятой:-0

Представления числа 0b01111111111111111111111111111111

Целочисленная беззнаковая интерпретация,16-ичное представление: 7fffffff

Целочисленная беззнаковая интерпретация,10-ичное представлении: 2147483647

Целочисленная знаковая интерпретация,16-ичное представление: 7fffffff

Целочисленная знаковая интерпретация,10-ичное представлении: 2147483647

Интерпретация в качестве числа с плавающей запятой:nan

[illegible]

Целочисленная беззнаковая интерпретация, 16-ичное представление:

ffffffffffffffff

Целочисленная беззнаковая интерпретация, 10-ичное представлении:

18446744073709551615

Целочисленная знаковая интерпретация, 16-ичное представление: ffffffffffffffff

Целочисленная знаковая интерпретация, 10-ичное представлении: -1

Интерпретация в качестве числа с плавающей запятой:nan

Представления числа

[illegible]

Целочисленная беззнаковая интерпретация, 16-ичное представление:

800000000000000000000000

Целочисленная беззнаковая интерпретация, 10-ичное представление:

9223372036854775808

Целочисленная знаковая интерпретация, 16-ичное представление: 8000000000000000

Целочисленная знаковая интерпретация, 10-ичное представление: -

9223372036854775808

Интерпретация в качестве числа с плавающей запятой: -0

Представления числа

[illegible]

Целочисленная беззнаковая интерпретация, 16-ичное представление:

7fffffffffffffffffffffffff

Целочисленная беззнаковая интерпретация, 10-ичное представлении:

9223372036854775807

Целочисленная знаковая интерпретация, 16-ичное представление: 7fffffffffffffff

Целочисленная знаковая интерпретация, 10-ичное представлении:

9223372036854775807

Интерпретация в качестве числа с плавающей запятой:nan

Представление числа 0xA1B2C3D4 как целочисленного:

Целочисленная беззнаковая интерпретация, 16-ичное представление: a1b2c3d4

Целочисленная беззнаковая интерпретация, 10-ичное представлении: 2712847316

Целочисленная знаковая интерпретация, 16-ичное представление: a1b2c3d4

Целочисленная знаковая интерпретация, 10-ичное представлении: 2712847316

Интерпретация в качестве числа с плавающей запятой: 1.34032e-314

Представление числа 3 как целочисленного:

Целочисленная беззнаковая интерпретация, 16-ичное представление: 3

Целочисленная беззнаковая интерпретация, 10-ичное представление: 3

Целочисленная знаковая интерпретация, 16-ичное представление: 3

Целочисленная знаковая интерпретация, 10-ичное представлении: 3

Интерпретация в качестве числа с плавающей запятой:1.4822e-323

Представление числа -8 как целочисленного:

Целочисленная беззнаковая интерпретация, 16-ичное представление:

```
ffffffffff8
```

Целочисленная беззнаковая интерпретация,10-ичное представлении:
18446744073709551608
Целочисленная знаковая интерпретация,16-ичное представление: ffffffffffffffff8
Целочисленная знаковая интерпретация,10-ичное представлении: -8
Интерпретация в качестве числа с плавающей запятой:nan

Представление числа 0xA1B2C3D4 с плавающей запятой:
Целочисленная беззнаковая интерпретация,16-ичное представление:
41e436587a800000
Целочисленная беззнаковая интерпретация,10-ичное представлении:
4747979660795576320
Целочисленная знаковая интерпретация,16-ичное представление: 41e436587a800000
Целочисленная знаковая интерпретация,10-ичное представлении:
4747979660795576320
Интерпретация в качестве числа с плавающей запятой:2.71285e+09

Представление числа 3 с плавающей запятой:
Целочисленная беззнаковая интерпретация,16-ичное представление:
4008000000000000
Целочисленная беззнаковая интерпретация,10-ичное представлении:
4613937818241073152
Целочисленная знаковая интерпретация,16-ичное представление: 4008000000000000
Целочисленная знаковая интерпретация,10-ичное представлении:
4613937818241073152
Интерпретация в качестве числа с плавающей запятой:3

Представление числа -8 с плавающей запятой:
Целочисленная беззнаковая интерпретация,16-ичное представление:
c020000000000000
Целочисленная беззнаковая интерпретация,10-ичное представлении:
13844065254536904704
Целочисленная знаковая интерпретация,16-ичное представление: c020000000000000
Целочисленная знаковая интерпретация,10-ичное представлении: -
4602678819172646912
Интерпретация в качестве числа с плавающей запятой:-8

bin	unsigned hex	unsigned dec	signed hex	signed dec	float
00000000000000000000000000000000 000	0	0	0	0	0
11111111111111111111111111111111 111	ffffffff	4294967295	ffffffff	-1	-nan
10000000000000000000000000000000 000	80000000	2147483648	80000000	-2147483648	-0
01111111111111111111111111111111 111	7fffffff	2147483647	7fffffff	2147483647	nan
10100001101100101100001111010 100	a1b2c3d4	2712847316	a1b2c3d4	-1582119980	-1.21136e-18

00000000000000000000000000000000 011	3	3	3	3	4.2039e-45
11111111111111111111111111111111 000	ffffff8	4294967288	ffffff8	-8	-nan
01001111001000011011001011000 100	4f21b2c4	1327608516	4f21b2c4	1327608516	2.71285e+09
01000000010000000000000000000000 000	40400000	1077936128	40400000	1077936128	3
11000001000000000000000000000000 000	c1000000	3238002688	c1000000	-1056964608	-8

bin	unsigned hex	unsigned dec	signed hex	signed dec	double
00000000000000000000000000000000 00000000000000000000000000000000 000000	0	0	0	0	0
11111111111111111111111111111111 11111111111111111111111111111111 111111	ffffffffffffff f	184467440737 09551615	ffffffffffff ffff	-1	-nan
10000000000000000000000000000000 00000000000000000000000000000000 000000	80000000000000 0	922337203685 4775808	800000000000 0000	- 9223372036 854775808	-0
01111111111111111111111111111111 11111111111111111111111111111111 111111	7ffffffffffffff f	922337203685 4775807	7ffffffffffff ffff	9223372036 854775807	nan
00000000000000000000000000000000 00010100001101100101100001111 010100	a1b2c3d4	2712847316	a1b2c3d4	2712847316	1.34032e-314
00000000000000000000000000000000 00000000000000000000000000000000 000011	3	3	3	3	1.4822e-323
11111111111111111111111111111111 11111111111111111111111111111111 111000	ffffffffffffff 8	184467440737 09551608	ffffffffffff fff8	-8	-nan
01000001111001000011011001011 00001111010100000000000000000000 000000	41e436587a80000 0	474797966079 5576320	41e436587a80 0000	4747979660 795576320	2.71285e+09
01000000000010000000000000000000 00000000000000000000000000000000 000000	40080000000000 0	461393781824 1073152	400800000000 0000	4613937818 241073152	3
11000000001000000000000000000000 00000000000000000000000000000000 000000	c0200000000000 0	138440652545 36904704	c02000000000 0000	- 4602678819 172646912	-8

Задание 1.4

Задание:

Разработайте программу на языке C++, которая расширяет значение целочисленной переменной из 16 бит до 32 бит, рассматривая числа как:

- знаковые (*signed*);
- беззнаковые (*unsigned*).

Проверьте её работу на значениях *y* и *z* (таблица Л1.1). Исходное значение, а также результаты беззнакового и знакового расширения должны печататься в представлениях (а)–(г) (функциями *print16()* и *print32()*). Полученные результаты внесите в отчёт.

Листинг:

```
void CastSignedFrom16To32(short p)
{
    int signed_res32 = static_cast<int> (p);
    cout<<"Расширение 2 -входная переменная рассматривается знаковой"<<endl;
    print32(&signed_res32);
}

void CastUnsignedFrom16To32(unsigned short p)
{
    unsigned int unsigned_res32 = static_cast<unsigned int> (p);
    cout<<"Расширение 1 -входная переменная рассматривается беззнаковой"<<endl;
    print32(&unsigned_res32);
}

int main()
{
    setlocale(0, "");
    cout<<"-----Задание 4-----"<<endl;
    short y__ = 3;
    short z__ = -8;
    cout<<"Входное значение:"<<endl;
    print16(&y__);
    CastUnsignedFrom16To32(y__);
    CastSignedFrom16To32(y__);

    cout<<"Входное значение:"<<endl;
    print16(&z__);
    CastUnsignedFrom16To32(z__);
    CastSignedFrom16To32(z__);
}
```

Пример вывода:

-----Задание 4-----

Входное значение:

Целочисленная беззнаковая интерпретация,16-ичное представление: 3

Целочисленная беззнаковая интерпретация,10-ичное представлении: 3
 Целочисленная знаковая интерпретация,16-ичное представление: 3
 Целочисленная знаковая интерпретация,10-ичное представлении: 3

Расширение 1 -входная переменная рассматривается беззнаковой

Целочисленная беззнаковая интерпретация,16-ичное представление: 3
 Целочисленная беззнаковая интерпретация,10-ичное представлении: 3
 Целочисленная знаковая интерпретация,16-ичное представление: 3
 Целочисленная знаковая интерпретация,10-ичное представлении: 3
 Интерпретация в качестве числа с плавающей запятой:4.2039e-45

Расширение 2 -входная переменная рассматривается знаковой

Целочисленная беззнаковая интерпретация,16-ичное представление: 3
 Целочисленная беззнаковая интерпретация,10-ичное представлении: 3
 Целочисленная знаковая интерпретация,16-ичное представление: 3
 Целочисленная знаковая интерпретация,10-ичное представлении: 3
 Интерпретация в качестве числа с плавающей запятой:4.2039e-45

Входное значение:

Целочисленная беззнаковая интерпретация,16-ичное представление: fff8
 Целочисленная беззнаковая интерпретация,10-ичное представлении: 65528
 Целочисленная знаковая интерпретация,16-ичное представление: fff8
 Целочисленная знаковая интерпретация,10-ичное представлении: -8

Расширение 1 -входная переменная рассматривается беззнаковой

Целочисленная беззнаковая интерпретация,16-ичное представление: fff8
 Целочисленная беззнаковая интерпретация,10-ичное представлении: 65528
 Целочисленная знаковая интерпретация,16-ичное представление: fff8
 Целочисленная знаковая интерпретация,10-ичное представлении: 65528
 Интерпретация в качестве числа с плавающей запятой:9.18243e-41

Расширение 2 -входная переменная рассматривается знаковой

Целочисленная беззнаковая интерпретация,16-ичное представление: ffffffff8
 Целочисленная беззнаковая интерпретация,10-ичное представлении: 4294967288
 Целочисленная знаковая интерпретация,16-ичное представление: ffffffff8
 Целочисленная знаковая интерпретация,10-ичное представлении: -8
 Интерпретация в качестве числа с плавающей запятой:nan

in put	sh or t	ush ort	i n t	ui nt	he x1 6	hex3 2	bin16	bin32	uhe x16	uhe x32	ubin16	ubin32
3	3	3	3	3	3	3	00000000 00000011	0000000000000000 0000000000000011	3	3	00000000 00000011	0000000000000000 0000000000000011
-8	-8	655 28	- 8	65 52 8	ff f8	ffff fff8	11111111 11111000	1111111111111111 1111111111111000	fff 8	fff 8	11111111 11111000	0000000000000000 1111111111111000

Задание 1.5

Задача:

Изучите, как располагаются в памяти байты, составляющие число или структуру данных. 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).

С помощью `printDump()` определите и выпишите в отчёт, как хранятся в памяти компьютера в программе на C/C++:

- целое число x (типа `int`; таблица Л1.1); по результату исследования определите порядок следования байтов в словах для вашего процессора:
 - а) прямой (младший байт по младшему адресу, порядок Intel, Little-Endian, от младшего к старшему);
 - б) обратный (младший байт по старшему адресу, порядок Motorola, BigEndian, от старшего к младшему);
- массив из трёх целых чисел (статический или динамический, но не высокоуровневый контейнер) с элементами x , y , z ;
- число с плавающей запятой y (типа `double`; таблица Л1.1);
- строки "bcdxy" и "ёжзий" (массив из `char`; при выборе N учитывайте всю длину строки, а не только видимые буквы);
- «широкие» строки L"bcdxy" и L"ёжзий" (массив из `wchar_t`; при выборе N учитывайте всю длину строки).

Листинг:

```
void printDump(void* p, size_t N)
{
    cout << hex;
    auto pb = (uint8_t*)p;
    for(size_t i = 0; i < N; ++i)
    {
        cout << (int)pb[i] << " ";
    }
    cout << endl << dec;
}

int main()
{
```

```

cout<<"-----Задание 4-----"<<endl;
int x = 0xA1B2C3D4;
cout <<"Целое число" <<x<< endl;
printDump(&x, sizeof(int));
auto arr = new int[3]{(int)0xA1B2C3D4, 3, -8};
cout <<"Массив" << endl;
printDump(arr, sizeof(int) * 3);
delete[] arr;
double _y = 3;
cout <<"Число с плавающей запятой" << endl;
printDump(&_y, sizeof(double));
cout <<"Строки" << endl;
char str1[] = "bcdxy";
printDump(&str1, sizeof(char) * 6); // 5 chars + 0
char str2[] = "ёжик";
printDump(&str2, sizeof(char) * 11);
cout <<"Широкие строки" << endl;
wchar_t wstr1[] = L"bcdxy";
printDump(&wstr1, sizeof(wchar_t) * 6);
wchar_t wstr2[] = L"ёжик";
printDump(&wstr2, sizeof(wchar_t) * 6);
}

```

Пример вывода:

```

-----Задание 4-----
Целое число
d4 c3 b2 a1
Массив
d4 c3 b2 a1 3 0 0 0 f8 ff ff ff
Число с плавающей запятой
0 0 0 0 0 0 8 40
Строки
62 63 64 78 79 0
b8 e6 e7 e8 e9 0 62 63 64 78 79
Широкие строки
62 0 63 0 64 0 78 0 79 0 0 0
51 4 0 0 36 4 0 0 37 4 0 0 38 4 0 0 39 4 0 0 0 0 0 0

```