Лабораторная работа № 1:

“Представление данных в ЭВМ”

МИЭТ

Выполнили:

студенты группы МП-30

Алимагадов К. А., Карпухин Г. К.

Вариант № 9

Зеленоград 2018 год

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 шестнадцатиричной 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 шестналцатиричное представления шестнадцатибитных чисел x и y, а также беззнаковую интерпретацию этого представления в десятичном виде.

Вариант №1

x = -34; y = 2^{15} + 7

```
//программа выводит данные числа в разных системах счисления, а также их знаковую и беззнаковую интерпретации
#include "cmath"
#include <iostream>
#include <bitset>
using namespace std;
int main()
{
	unsigned short x1 = -34;
	short x2 = -34;

	unsigned short y1 = 32775;
	short y2 = 32775;


	cout << "signed x = " << x2 << endl;
	cout << bitset<16>(x2) << endl;
	cout << hex << x2 << endl;


	cout << "unsigned x = " << dec << x1 << endl;
	cout << bitset<16>(x1) << endl;
	cout << hex << x1 << endl;


	cout << "signed y = " << dec << y2 << endl;
	cout << bitset<16>(y2) << endl;
	cout << hex << y2 << endl;


	cout << "unsigned y= " << dec << y1 << endl;
	cout << bitset<16>(y1) << endl;
	cout << hex << y1 << endl;


	return 0;
}
```

```
//результаты работы программы:
```

X = -34 десятичное знаковое

X = 1111111111011110 двоичное

X = FFDE шестнадцатеричное

X = 65502 десятичное беззнаковое

Y = 215 + 7 = -32761 десятичное знаковое

Y = 1000000000000111 двоичное

Y = 8007 шестнадцатеричное

Y = 32775 десятичное беззнаковое

Беззнаковая интерпретация переменной в шестнадцатеричной форме совпадает со знаковой интерпретацией в шестнадцатеричной форме.

Беззнаковое интерпретация переменной в десятичной форме не совпадает со знаковой, т. к. при знаковой интерпретации старший бит отвечает за знак числа, а в беззнаковой отвечает за мантиссу.

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).

```
// программа выводит на экран значения минимального и максимального 16-битное числа со знаком и без знака в формах представления (а), (б), (в)

#include <bitset>
#include "cmath"
#include <iostream>
using namespace std;

int main()
{
	short min = SHRT_MIN;
	unsigned short usmin = min;
	short max = SHRT_MAX;
	unsigned short usmax = max;
	unsigned short umax = USHRT_MAX;
	short sumax = umax;
	unsigned short umin = 0;
	short sumin = umin;

	cout << "SHRT_MIN" << endl
		<< "DEC " << dec << min << endl
		<< "UDEC " << dec << usmin << endl
		<< "HEX " << hex << min << endl
		<< "BIN " << bitset<16>(min) << endl << endl;
	cout << "SHRT_MAX" << endl
		<< "DEC " << dec << max << endl
		<< "UDEC " << dec << usmax << endl
		<< "HEX " << hex << max << endl
```

```
            << "BIN " << bitset<16>(max) << endl << endl;
      cout << "USHRT_MIN" << endl
            << "DEC " << dec << sumin << endl
            << "UDEC " << dec << umin << endl
            << "HEX " << hex << umin << endl
            << "BIN " << bitset<16>(umin) << endl << endl;
      cout << "USHRT_MAX" << endl
            << "DEC " << dec << sumax << endl
            << "UDEC " << dec << umax << endl
            << "HEX " << hex << umax << endl
            << "BIN " << bitset<16>(umax) << endl << endl;

}

//результаты работы программы:
```

Минимальное 16-битное знаковое число в:

десятичной знаковой форме (-32768);

десятичной беззнаковой форме (32768);

шестнадцатиричной беззнаковой форме (8000);

двоичной форме (1000000000000000).

Максимальное 16-битное знаковое число в:

десятичной знаковой форме (32767);

десятичной беззнаковой форме (32767);

шестнадцатиричной беззнаковой форме (7fff);

двоичной форме (0111111111111111).

Минимальное 16-битное беззнаковое число в:

десятичной знаковой форме (0);

десятичной беззнаковой форме (0);

шестнадцатиричной беззнаковой форме (0);

двоичной форме (0000000000000000).

Максимальное 16-битное беззнаковое число в:

десятичной знаковой форме (-1);

десятичной беззнаковой форме (65535);

шестнадцатиричной беззнаковой форме (ffff);

двоичной форме (1111111111111111).

Задание № 3

Разработайте программу на языке C++, выполняющую над беззнаковым шестнадцатибитными целыми числами следующие поразрядные операции (результат должен печататься в десятичной и шестнадцатиричной формах): конъюнкция, дизъюнкция, сложение по модулю 2, отрицание, дополнение до двух, логический сдвиг влево, логический сдвиг вправо.

Вариант № 3

x = 0x9211
y = 0x0004

x2 = 0x0009
y2 = 0x0013

```
// данная программа осуществляет все вышеперечисленные операции и выводит результат
#include "cmath"
#include <iostream>
#include <fstream>
#include <bitset>
using namespace std;
ifstream vi;
ofstream vv;


template <typename T>
T Conjuction(T & const first, T & const second)
{
	T temp = first & second;
	vv << "Conjuction" << endl;
	vv << "Dec interpretation: " << dec << (unsigned short)temp << endl;
	vv << "Hex interpretation: " << hex << (unsigned short)temp << endl;
	return temp;
}

template <typename T>
T Disjunction(T & const first, T & const second)
{
	T temp = first | second;
	vv << "Disjunction" << endl;
	vv << "Dec interpretation: " << dec << (unsigned short)temp << endl;
	vv << "Hex interpretation: " << hex << (unsigned short)temp << endl;
	return temp;
}

template <typename T>
T Addition(T & const first, T & const second)
{
	T temp = first ^ second;
	vv << "Addition" << endl;
	vv << "Dec interpretation: " << dec << (unsigned short)temp << endl;
	vv << "Hex interpretation: " << hex << (unsigned short)temp << endl;
	return temp;
}
```

```
template <typename T>
T Negation(T & const exmp)
{
	T temp = ~exmp;
	vv << "Negation" << endl;
	vv << "Dec interpretation: " << dec << (unsigned short)temp << endl;
	vv << "Hex interpretation: " << hex << (unsigned short)temp << endl;
	return temp;
}

template <typename T>
T Add_to_two(T & const exmp)
{
	T temp = (~exmp) | exmp;
	vv << "Add_to_two" << endl;
	vv << "Dec interpretation: " << dec << (unsigned short)temp << endl;
	vv << "Hex interpretation: " << hex << (unsigned short)temp << endl;
	return temp;
}

template <typename T>
T Shift_to_left(T & const exmp, T & const count)
{
	T temp = exmp << count;
	vv << "Shift_to_left" << endl;
	vv << "Dec interpretation: " << dec << (unsigned short)temp << endl;
	vv << "Hex interpretation: " << hex << (unsigned short)temp << endl;
	return temp;
}

template <typename T>
T Shift_to_right(T & const exmp, T & const count)
{
	T temp = exmp >> count;
	vv << "Shift_to_right" << endl;
	vv << "Dec interpretation: " << dec << (unsigned short)temp << endl;
	vv << "Hex interpretation: " << hex << (unsigned short)temp << endl;
	return temp;
}

int main()
{
	vv.open("res.txt");
	unsigned short x = 0x9211;
	unsigned short y = 0x0004;
	vv << "x = " << x << endl;
	vv << "y = " << y << endl;
	Conjuction<unsigned short>(x, y);
	Disjunction<unsigned short>(x, y);
	Addition<unsigned short>(x, y);
	Negation<unsigned short>(x);
	Negation<unsigned short>(y);
	Add_to_two<unsigned short>(x);
	Add_to_two<unsigned short>(y);
	Shift_to_left<unsigned short>(x, y);
	Shift_to_right<unsigned short>(x, y);
	vv << endl << endl;
	unsigned short x2 = 0x0009;
	unsigned short y2 = 0x0013;
	vv << "x2 = " << x2 << endl;
	vv << "y2 = " << y2 << endl;
	Conjuction<unsigned short>(x2, y2);
	Disjunction<unsigned short>(x2, y2);
	Addition<unsigned short>(x2, y2);
	Negation<unsigned short>(x2);
```

```
        Negation<unsigned short>(y2);
        Add_to_two<unsigned short>(x2);
        Add_to_two<unsigned short>(y2);
        Shift_to_left<unsigned short>(x2, y2);
        Shift_to_right<unsigned short>(x2, y2);
        vv.close();
        return 0;
}

//результаты работы программы:
```
x = 37393

y = 4

Conjuction

Dec interpretation: 0

Hex interpretation: 0

Disjunction

Dec interpretation: 37397

Hex interpretation: 9215

Addition

Dec interpretation: 37397

Hex interpretation: 9215

Negation

Dec interpretation: 28142

Hex interpretation: 6dee

Negation

Dec interpretation: 65531

Hex interpretation: fffb

Add_to_two

Dec interpretation: 65535

Hex interpretation: ffff

Add_to_two

Dec interpretation: 65535

Hex interpretation: ffff

Shift_to_left

Dec interpretation: 8464

Hex interpretation: 2110

Shift_to_right

Dec interpretation: 2337

Hex interpretation: 921

x2 = 9

y2 = 13

Conjuction

Dec interpretation: 1

Hex interpretation: 1

Disjunction

Dec interpretation: 27

Hex interpretation: 1b

Addition

Dec interpretation: 26

Hex interpretation: 1a

Negation

Dec interpretation: 65526

Hex interpretation: fff6

Negation

Dec interpretation: 65516

Hex interpretation: ffec

Add_to_two

Dec interpretation: 65535

Hex interpretation: ffff

Add_to_two

Dec interpretation: 65535

Hex interpretation: ffff

Shift_to_left

Dec interpretation: 0

Hex interpretation: 0

Shift_to_right

Dec interpretation: 0

Hex interpretation: 0

Задание № 4

Измените в программе из задания 3 тип переменных на знаковый. Объясните изменение или неизменность результата.

```
int main()
{
	vv.open("res.txt");
	unsigned char x = 0x9211;
	unsigned char y = 0x0004;
	vv << "x = " << (unsigned short)x << endl;
	vv << "y = " << (unsigned short)y << endl;
	Conjuction<unsigned char>(x, y);
	Disjunction<unsigned char>(x, y);
	Addition<unsigned char>(x, y);
	Negation<unsigned char>(x);
	Negation<unsigned char>(y);
	Add_to_two<unsigned char>(x);
	Add_to_two<unsigned char>(y);
	Shift_to_left<unsigned char>(x, y);
	Shift_to_right<unsigned char>(x, y);
	unsigned char x2 = 0x0009;
	unsigned char y2 = 0x0013;
	vv << "x2 = " << (unsigned short)x2 << endl;
	vv << "y2 = " << (unsigned short)y2 << endl;
	Conjuction<unsigned char>(x2, y2);
	Disjunction<unsigned char>(x2, y2);
	Addition<unsigned char>(x2, y2);
	Negation<unsigned char>(x2);
	Negation<unsigned char>(y2);
	Add_to_two<unsigned char>(x2);
	Add_to_two<unsigned char>(y2);
	Shift_to_left<unsigned char>(x2, y2);
	Shift_to_right<unsigned char>(x2, y2);
	vv.close();
	return 0;
}

//результаты работы программы:
```

x = 17

y = 4

Conjuction

Dec interpretation: 0

Hex interpretation: 0

Disjunction

Dec interpretation: 21

Hex interpretation: 15

Addition

Dec interpretation: 21

Hex interpretation: 15

Negation

Dec interpretation: 238

Hex interpretation: ee

Negation

Dec interpretation: 251

Hex interpretation: fb

Add_to_two

Dec interpretation: 255

Hex interpretation: ff

Add_to_two

Dec interpretation: 255

Hex interpretation: ff

Shift_to_left

Dec interpretation: 16

Hex interpretation: 10

Shift_to_right

Dec interpretation: 1

Hex interpretation: 1

x2 = 9

y2 = 13

Conjuction

Dec interpretation: 1

Hex interpretation: 1

Disjunction

Dec interpretation: 27

Hex interpretation: 1b

Addition

Dec interpretation: 26

Hex interpretation: 1a

Negation

Dec interpretation: 246

Hex interpretation: f6

Negation

Dec interpretation: 236

Hex interpretation: ec

Add_to_two

Dec interpretation: 255

Hex interpretation: ff

Add_to_two

Dec interpretation: 255

Hex interpretation: ff

Shift_to_left

Dec interpretation: 0

Hex interpretation: 0

Shift_to_right

Dec interpretation: 0

Hex interpretation: 0

Размер unsigned char равен 1 байту, в то время у как unsigned short 2 байта, т. к. при записи в ячейку с меньшим объёмом памяти, часть информации отбросилась, то следовательно значения x и y в данном случае отличаются от предыдущих, а значит и результаты логческих операций над ними будут отличаться от прежних.

Задание № 5

Разработайте программу на языке C++ (или дополните программу из задания 3), которая расширяет шестнадцатибитное представление числа x до тридцатидвухбитного, рассматривая числа как

- знаковые(signed);

- беззнаковые(unsigned).

```
// данная программа приводит знаковые и беззнаковые шестнадцатибитовые числа к знаковым и беззнаковым тридцатидвухбитовым числам
#include <iostream>
#include "cmath"
#include <bitset>
using namespace std;


template <typename T>
int Representation_signed(T & const exmp)
{
	return (int)exmp;
}

template <typename T>
unsigned int Representation_unsigned(T & const exmp)
{
	return (unsigned int)exmp;
}

ifstream vi;
ofstream vv;


int main()
{
	vv.open("res.txt");
	short x = 0xABCD;
	vv << "x = " << dec << x << endl;
	vv << "Signed representation" << endl;
	vv << "Dec interpretation: " << dec << Representation_signed<short>(x) << endl;
	vv << "Hex interpretation: " << hex << Representation_signed<short>(x) << endl;
	vv << "Unsigned representation" << endl;
	vv << "Dec interpretation: " << dec << Representation_unsigned<short>(x) << endl;
	vv << "Hex interpretation: " << hex << Representation_unsigned<short>(x) << endl;
```

```
	unsigned short y = 0xABCD;
	vv << "y = " << dec << y << endl;
	vv << "Signed representation" << endl;
	vv << "Dec interpretation: " << dec << Representation_signed<unsigned short>(y) <<
endl;
	vv << "Hex interpretation: " << hex << Representation_signed<unsigned short>(y) <<
endl;
	vv << "Unsigned representation" << endl;
	vv << "Dec interpretation: " << dec << Representation_unsigned<unsigned short>(y)
<< endl;
	vv << "Hex interpretation: " << hex << Representation_unsigned<unsigned short>(y)
<< endl;
	vv.close();
	return 0;
}

//результаты работы программы:
```

x = -21555

Signed representation

Dec interpretation: -21555

Hex interpretation: ffffabcd

Unsigned representation

Dec interpretation: 4294945741

Hex interpretation: ffffabcd

y = 43981

Signed representation

Dec interpretation: 43981

Hex interpretation: abcd

Unsigned representation

Dec interpretation: 43981

Hex interpretation: abcd

Задание №6

Напишите программу, демонстрирующую переполнение целых чисел со знаком и без знака. Для этого опишите целое число со знаком и инициализируйте его максимально возможным для данного типа значением, результат распечатайте в шестнадцатиричном виде. Затем прибавьте к этому числу едиеницу, результат распечатайте также в

шестнадцатиричном виде. Выполните аналогичные действия для беззнаковой переменной. Выпишите в отчёт распечатанные значения переменной, поясните полученные результаты.

```
//в данной программе осуществляется переполнение целой переменной
int main()
{
	vv.open("res.txt");
	int x = INT64_MAX;
	vv << "Signed representation" << endl;
	vv << "Hex interpretation INT64_MAX: " << hex << x << endl;
	vv << "Hex interpretation (INT64_MAX + 1): " << hex << ++x << endl;
	unsigned int y = UINT64_MAX;
	vv << "Unsigned representation" << endl;
	vv << "Hex interpretation UINT64_MAX: " << hex << y << endl;
	vv << "Hex interpretation (UINT64_MAX + 1): " << hex << ++y << endl;
	vv.close();
	return 0;
}

//результаты работы программы:
```

Signed representation

Hex interpretation INT64_MAX: ffffffff

Hex interpretation (INT64_MAX + 1): 0

Unsigned representation

Hex interpretation UINT64_MAX: ffffffff

Hex interpretation (UINT64_MAX + 1): 0

Так как в данной задаче мы переполняем значение целой переменной, она принимает своё самое минимальное значение (в шестнадцатиричном формате вывода).

Задание №7

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);

- строки “abcd” и “абвг” (массив из char);

- широкие строки L “abcd” b L “абвг” (массив из wchar_t):

Целое двухбайтовое число 0x12345678 при прямом порядке (Little-Endian) имеет в памяти вид: 78 56 34 12

```
//данные программы выводит адреса данных в памяти
int main()
{
	char tmp[4] = { 'a','b','c','d' };
	for (int i = 0; i < 4; i++)
	{
		cout << hex << (int)tmp[i] << " ";
	}
	return 0;
}
```

Строка “abcd” имеет в памяти вид: 61 62 63 64

```
int main()
{
	char tmp[4] = { 'a','б','в','г' };
	for (int i = 0; i < 4; i++)
	{
		cout << hex << (int)tmp[i] << " ";
	}
	return 0;
}
```

Строка “абвг” имеет в памяти вид: ffffffe0 ffffffe1 ffffffe2 ffffffe3

```
int main()
{
	wchar_t tmp[4] = { 'a','b','c','d' };
	for (int i = 0; i < 4; i++)
	{
		cout << hex << (int)tmp[i] << " ";
	}
	return 0;
}
```

Строка L “abcd” имеет в памяти вид: 61 62 63 64

```
int main()
{
	wchar_t tmp[4] = { 'a','б','в','г' };
	for (int i = 0; i < 4; i++)
	{
		cout << hex << (int)tmp[i] << " ";
	}
	return 0;
}
```

Строка L “абвг” имеет в памяти вид: ffe0 ffe1 ffe2 ffe3

Задание №8-9 При помощи оператора sizeof выясните, сколько байтов занимают переменные следуюзих типов: char, bool, wchar_t, short, int, long, long long, float, double, long double, size_t, ptrdiff_t, void*.

Результаты формите в отчёте в виде таблицы, указывая для каждого типа его назначение.

Запустите программы из заданий 7-8 на двух других платформах, доступных на ВЦ – 32 и 64 разрядных версиях Microsoft Windows и повторите измерения.

```
//данная программа определяет размеры типов данных
#include "stdafx.h"
#include "iostream"
#include <cstddef>
using namespace std;


template<typename T>
void Size()
{
	cout << sizeof(T) << endl;
}

int main()
{
	Size<char>();
	Size<bool>();
	Size<wchar_t>();
	Size<short>();
	Size<int>();
	Size<long>();
	Size<long long>();
	Size<float>();
	Size<double>();
	Size<long double>();
	Size<size_t>();
	Size<ptrdiff_t>();
	Size<void*>();
    return 0;
}
//результаты работы программы:
```

Тип данных	Win32	Современный стандарт C++	Win64	Ubuntu32	Ubuntu64	Назначение
char	1	1	1	1	1	целочисленный (символьный) тип
bool	1	1	1	1	1	целочисленный (логический) тип
wchar_t	2	Не меньше, чем 2	2	4	4	unicode-тип символов
short	2	Не меньше, чем 2	2	2	2	целочисленный тип, размер которого больше или равен размеру типа меньше или равен размеру типа
int	4	Не меньше, чем 2	4	4	4	int — это целочисленный тип, размер которого больше или равен

						размеру типа меньше или равен размеру типа
long	4	Не меньше, чем 4	4	4	8	long (или long int целочисленный тип, размер которого больше или равен размеру типа
long long	8	Не меньше, чем 8	8	8	8	Больше, чем unsigned long
float	4	4	4	4	4	float — это тип с плавающей запятой наименьшего
double	8	Не меньше, чем 4	8	8	8	double — это тип с плавающей запятой, размер которого больше или равен размеру типа меньше или равен размеру типа long double
long double	8	Не меньше, чем 8	8	12	16	long double — это тип с плавающей запятой, размер которого больше или равен размеру типа
size_t	4	Не меньше, чем 2	8	4	8	целочисленный беззнаковый тип результата, возвращаемого операторами sizeof gnof
ptrdiff_t	4	Не меньше, чем 4	8	4	8	Указатель на неопределенный тип, этот указатель мы можем поместить любой тип.
void*	4	Не меньше, чем 4	8	4	8	базовый знаковый целочисленный тип, является типом результата выражения, где один указатель вычитается из другого