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 №7**

**Ассемблерные вставки в программах на C++. Флаги и условия**

Работу выполнил студенты группы ПИН-24 Баранов Д.А. и Демочкина А.В.

Работу проверил ассистент Института СПИНТех Фомин Р.А.

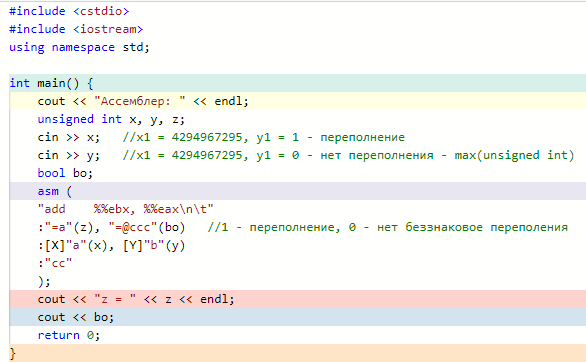
**Цель работы:** научиться анализировать флаги x86 и использовать условные команды.

**Задание 1.**

**Вариант 2**

Вычислите сумму двух целых чисел 𝑧 = 𝑥 + 𝑦. Определите корректность (отсутствие/наличие переполнения) результата, если трактовать его как

****

****

**#include <cstdio>**

**#include <iostream>**

**using namespace std;**

**int main() {**

**cout << "Ассемблер: " << endl;**

**unsigned int x, y, z;**

**cin >> x; //x1 = 4294967295, y1 = 1 - переполнение**

**cin >> y; //x1 = 4294967295, y1 = 0 - нет переполнения - max(unsigned int)**

**bool bo;**

**asm (**

**"add %%ebx, %%eax\n\t"**

**:"=a"(z), "=@ccc"(bo) //1 - переполнение, 0 - нет беззнаковое переполения**

**:[X]"a"(x), [Y]"b"(y)**

**:"cc"**

**);**

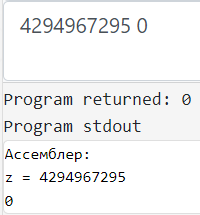
**cout << "z = " << z << endl;**

**cout << bo;**

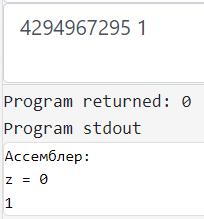
**return 0;**

**}**

Нет переполнения:



Переполнение:



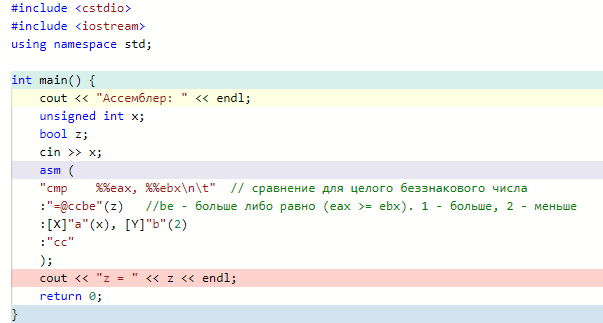
CC: c - значение регистра флага состояния CF (CF = 1 - беззнаковое переполнение)

**Задание 2.**

**Вариант 2**

Вычислите для заданного целого беззнакового 𝑥:





#include <cstdio>

#include <iostream>

using namespace std;

int main() {

cout << "Ассемблер: " << endl;

unsigned int x;

bool z;

cin >> x;

asm (

"cmp %%eax, %%ebx\n\t" // сравнение для целого беззнакового числа

:"=@ccbe"(z) //be - больше либо равно (eax >= ebx). 1 - больше, 2 - меньше

:[X]"a"(x), [Y]"b"(2)

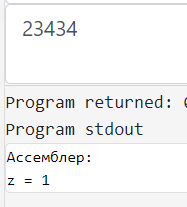
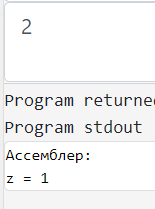
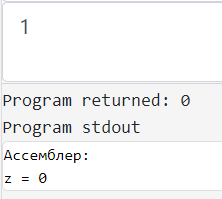
:"cc"

);

cout << "z = " << z << endl;

return 0;

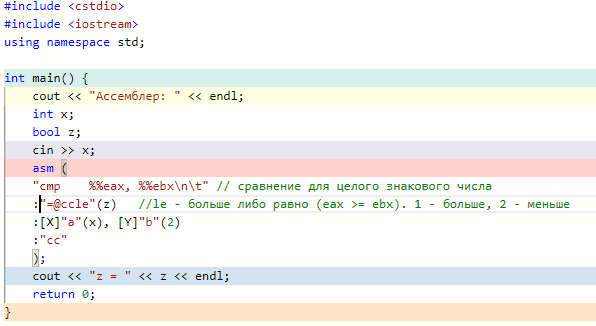
}



CC: be - значение регистра флагов состояния CF и ZF (CF = 1 и ZF = 1 - первое число больше либо равно второму как беззнаковое)

**Задание 3.**

Реализуйте Л7.з2 для целого знакового 𝑥.



#include <cstdio>

#include <iostream>

using namespace std;

int main() {

cout << "Ассемблер: " << endl;

int x;

bool z;

cin >> x;

asm (

"cmp %%eax, %%ebx\n\t" // сравнение для целого знакового числа

:"=@ccle"(z) //le - больше либо равно (eax >= ebx). 1 - больше, 2 - меньше

:[X]"a"(x), [Y]"b"(2)

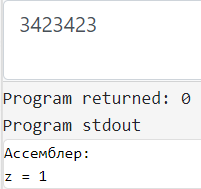
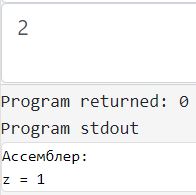
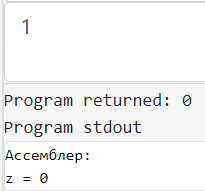
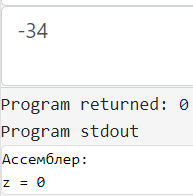
:"cc"

);

cout << "z = " << z << endl;

return 0;

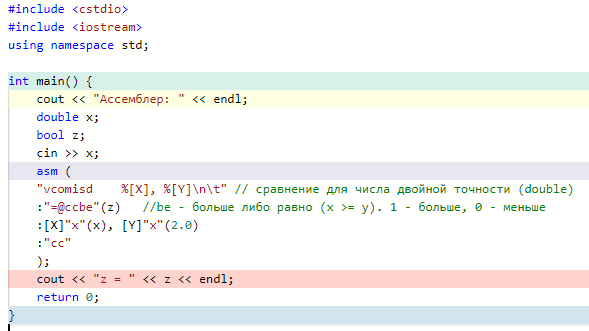
}



CC: le - значение регистра флагов состояния SF, OF и ZF (SF != OF и ZF = 1 - первое число больше либо равно второму как знаковое)

**Задание 4.**

Реализуйте Л7.з2 для 𝑥 с плавающей запятой двойной точности (𝑑𝑜𝑢𝑏𝑙𝑒), используя AVX-команду сравнения 𝑣𝑐𝑜𝑚𝑖𝑠𝑑 (или её SSE-аналог 𝑐𝑜𝑚𝑖𝑠𝑑).



#include <cstdio>

#include <iostream>

using namespace std;

int main() {

cout << "Ассемблер: " << endl;

double x;

bool z;

cin >> x;

asm (

"vcomisd %[X], %[Y]\n\t" // сравнение для числа двойной точности (double)

:"=@ccbe"(z) //be - больше либо равно (x >= y). 1 - больше, 0 - меньше

:[X]"x"(x), [Y]"x"(2.0)

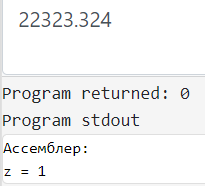
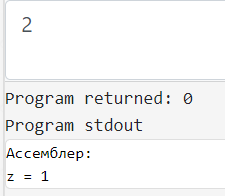
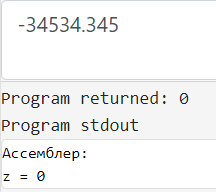
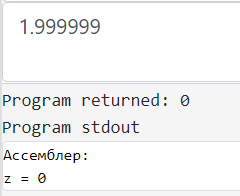
:"cc"

);

cout << "z = " << z << endl;

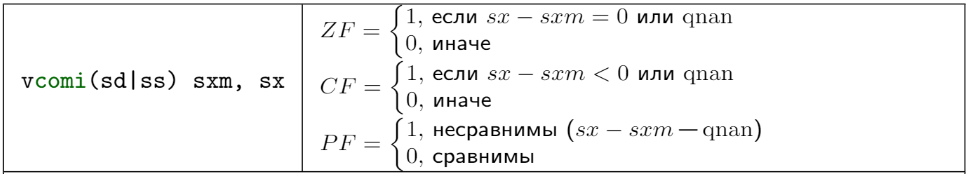
return 0;

}



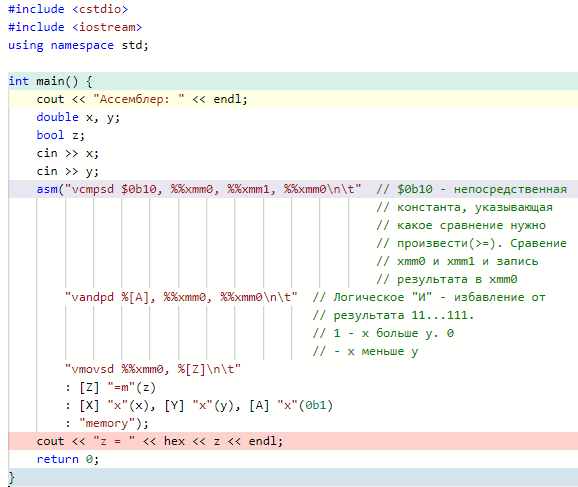
sd - двойная точность (double)

CC: be - значение регистра флагов состояния CF и ZF (CF = 1 и ZF = 1 - первое число больше либо равно второму как число двойной точности)



**Задание 5.**

Реализуйте Л7.з2 для 𝑥 с плавающей запятой двойной точности, используя AVX-команду сравнения 𝑣𝑐𝑚𝑝𝑠𝑑 (или её SSE-аналог) и битовые операции.



#include <cstdio>

#include <iostream>

using namespace std;

int main() {

cout << "Ассемблер: " << endl;

double x, y;

bool z;

cin >> x;

cin >> y;

asm("vcmpsd $0b10, %%xmm0, %%xmm1, %%xmm0\n\t" // $0b10 - непосредственная

// константа, указывающая

// какое сравнение нужно

// произвести(>=). Сравение

// xmm0 и xmm1 и запись

// результата в xmm0

"vandpd %[A], %%xmm0, %%xmm0\n\t" // Логическое "И" - избавление от

// результата 11...111.

// 1 - x больше y. 0

// - x меньше y

"vmovsd %%xmm0, %[Z]\n\t"

: [Z] "=m"(z)

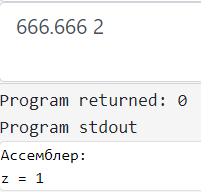
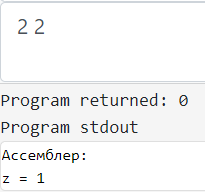
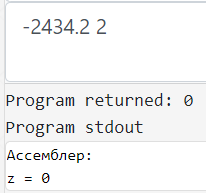
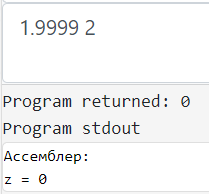
: [X] "x"(x), [Y] "x"(y), [A] "x"(0b1)

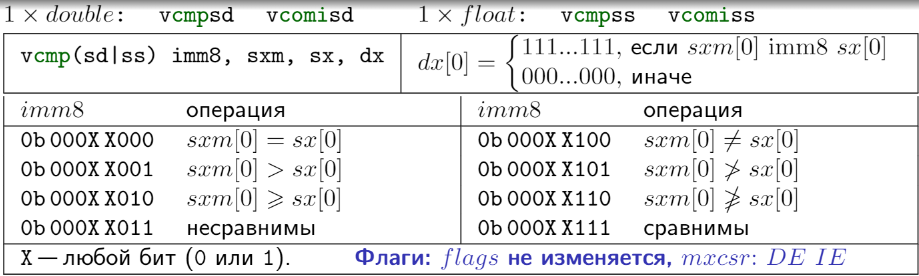
: "memory");

cout << "z = " << hex << z << endl;

return 0;

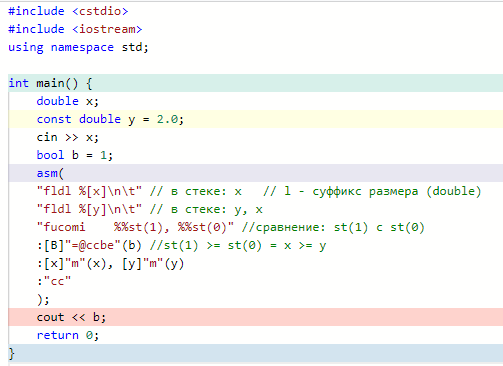
}





**Задание 6.**

Реализуйте Л7.з2 для 𝑥 с плавающей запятой двойной точности, используя FPU-команды сравнения 𝑓[𝑢]𝑐𝑜𝑚𝑖[𝑝].

****

**#include <cstdio>**

**#include <iostream>**

**using namespace std;**

**int main() {**

**double x;**

**const double y = 2.0;**

**cin >> x;**

**bool b = 1;**

**asm(**

**"fldl %[x]\n\t" // в стеке: x // l - cуффикс размера (double)**

**"fldl %[y]\n\t" // в стеке: y, x**

**"fucomi %%st(1), %%st(0)" //сравнение: st(1) c st(0)**

**:[B]"=@ccbe"(b) //st(1) >= st(0) = x >= y**

**:[x]"m"(x), [y]"m"(y)**

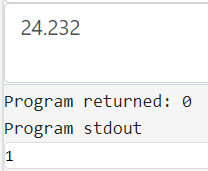
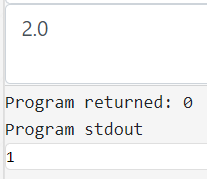
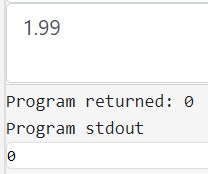
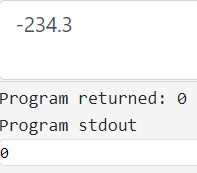
**:"cc"**

**);**

**cout << b;**

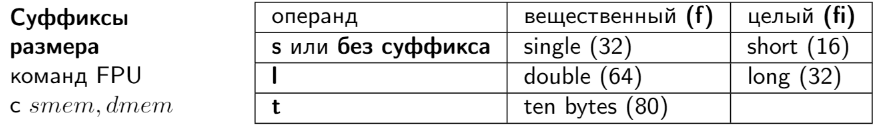
**return 0;**

**}**

****

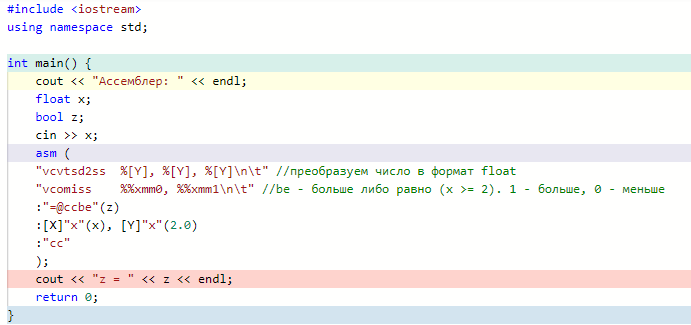






**Задание 7.**

**AVX:**

****

**#include <cstdio>**

**#include <iostream>**

**using namespace std;**

**int main() {**

**cout << "Ассемблер: " << endl;**

**float x;**

**bool z;**

**cin >> x;**

**asm (**

**"vcvtsd2ss %[Y], %[Y], %[Y]\n\t" //преобразуем число в формат float**

**"vcomiss %%xmm0, %%xmm1\n\t" //be - больше либо равно (x >= 2). 1 - больше, 0 - меньше**

**:"=@ccbe"(z)**

**:[X]"x"(x), [Y]"x"(2.0)**

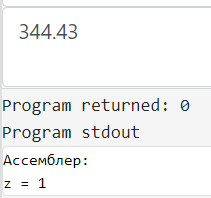
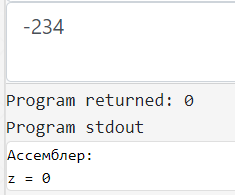
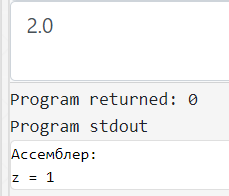
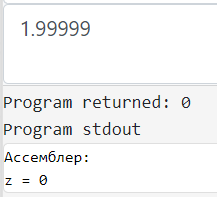
**:"cc"**

**);**

**cout << "z = " << z << endl;**

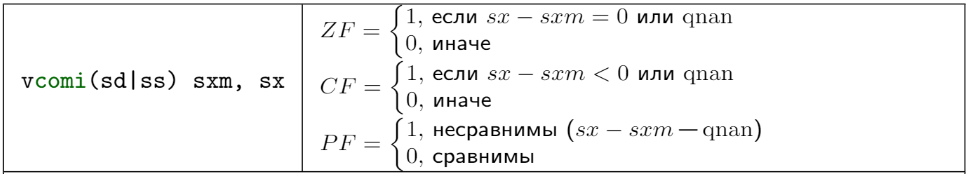
**return 0;**

**}**

****

ss -одинарная точность (float)

CC: be - значение регистра флагов состояния CF и ZF (CF = 1 и ZF = 1 - первое число больше либо равно второму как число одинарной точности)



**FPU:**

****

**#include <cstdio>**

**#include <iostream>**

**using namespace std;**

**int main() {**

**float x;**

**const float y = 2.0;**

**cin >> x;**

**bool b = 1;**

**asm(**

**"flds %[x]\n\t" // в стеке: x // s / отсутствие - cуффикс размера (float)**

**"flds %[y]\n\t" // в стеке: y, x**

**"fucomi %%st(1), %%st(0)" //сравнение: st(1) c st(0)**

**:[B]"=@ccbe"(b) //st(1) >= st(0) = x >= y**

**:[x]"m"(x), [y]"m"(y)**

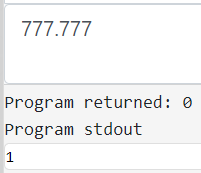
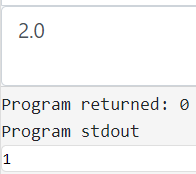
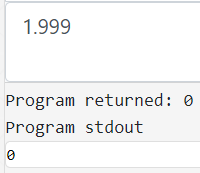
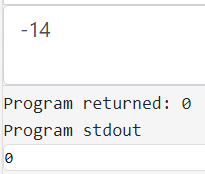
**:"cc"**

**);**

**cout << b;**

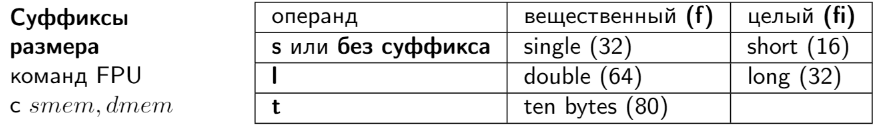
**return 0;**

**}**

****

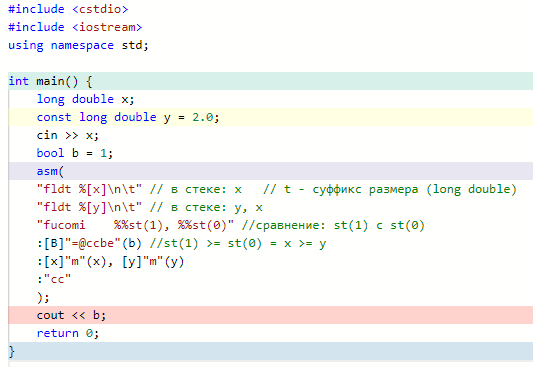






**Задание 8.**

Реализуйте Л7.з2 для 𝑥 с плавающей запятой двойной расширенной точности (𝑙𝑜𝑛𝑔 𝑑𝑜𝑢𝑏𝑙𝑒).

****

**#include <cstdio>**

**#include <iostream>**

**using namespace std;**

**int main() {**

**long double x;**

**const long double y = 2.0;**

**cin >> x;**

**bool b = 1;**

**asm(**

**"fldt %[x]\n\t" // в стеке: x // t - cуффикс размера (long double)**

**"fldt %[y]\n\t" // в стеке: y, x**

**"fucomi %%st(1), %%st(0)" //сравнение: st(1) c st(0)**

**:[B]"=@ccbe"(b) //st(1) >= st(0) = x >= y**

**:[x]"m"(x), [y]"m"(y)**

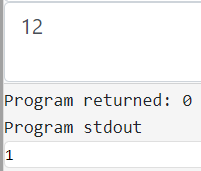
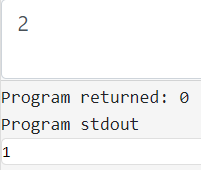
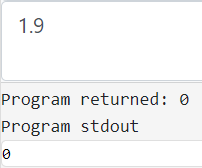
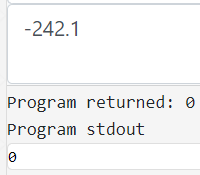
**:"cc"**

**);**

**cout << b;**

**return 0;**

**}**

****





