

## Сложение двух чисел с плавающей точкой

	<pre>%include 'io.inc'  section .data x dd 11.2 y dd 0.7  section .bss z resd 1  section .text global CMAIN CMAIN:     fld dword [x]     fld dword [y]     faddp     fstp dword [z]     PRINT_HEX 4, z     NEWLINE     xor eax, eax     ret</pre>
--	---

## Печать числа

	<pre>%include 'io.inc' section .data x dd 11.2 y dd 0.7 section .bss z resd 1 section .rodata lc db '%f', 10, 0 section .text CEXTERN printf global CMAIN CMAIN:     ; пролог функции     sub     esp, 20      fld     dword [x]     fld     dword [y]     faddp     fst     dword [z]     fstp    <b>qword</b> [esp + 4]     mov     dword [esp], lc     call    printf      add     esp, 20     ; эпилог функции</pre>
--	--

Пример демонстрирует важность порядка выполнения операций сложения (расстановки скобок). То есть при заданных  $x, y, z$   $((y + z) + x) \neq (y + (z + x))$ . В первом случае получится 3.14, во втором 0 за счёт выравнивания при сложении.

<pre> #include 'io.inc'  section .data x dq 3.14 y dq 1e50 z dq -1e50  section .bss r resq 1  section .rodata lc db '%f', 10, 0  section .text CEXTERN printf global CMAIN CMAIN: ; ... пролог функции sub    esp, 20  fld    qword [x] fld    qword [y] fld    qword [z] faddp faddp fst    qword [r] fstp   qword [esp + 4] mov    dword [esp], lc call   printf  add    esp, 20 ; ... эпилог функции </pre>	<pre> CMAIN: ; ... пролог функции sub    esp, 20  fld    qword [y] fld    qword [z] fld    qword [x] faddp faddp fst    qword [r] fstp   qword [esp + 4] mov    dword [esp], lc call   printf  add    esp, 20 ; ... эпилог функции </pre>
--	---

Переполнение при умножении. Программа аварийно завершится, так как было снято маскирование исключительных ситуаций в управляющем регистре сопроцессора x87

	<pre>section .data x dq 1e200 y dq 1e200 z dq 1e200  section .bss r resq 1 cw resw 1  section .rodata lc db '%lf', 10, 0  CMAIN: ; ... пролог функции sub    esp, 20 fstcw  word [cw] and    word [cw], 11111111_11000000b fldcw  word [cw]  fld    qword [x] fld    qword [z] fmulp fld    qword [y] fld    qword [z] fmulp fsubp ; вызов printf  add    esp, 20 ; ... эпилог функции</pre>
--	--

## Сравнение чисел

<pre>int isLe(double x, float y) {     return x &lt;= y; }</pre>	<pre>isLe:      push    ebp     mov     ebp, esp     fld     dword [ebp+16]     fld     qword [ebp+8]     fucompp ; st0 vs. st1     fnstsw  ax     sahf     setbe  al     movzx  eax, al     pop    ebp     ret</pre>
--	---

## Функции: возвращение числа с плавающей точкой

<pre>void caller(double *p) {     *p = inverse(*p); }</pre>	<pre>caller:      push    ebp     mov     ebp, esp     sub     esp, 8     mov     eax, dword [ebp+8]     fld     dword [eax]     fstp    qword [esp]     call   inverse     mov     eax, dword [ebp+8]     fstp    qword [eax]     leave     ret</pre>
---	--

<pre>float inverse(double x) {     return 1/x; }</pre>	<pre>inverse:      push    ebp     mov     ebp, esp     fld1     fld     qword [ebp+8]     fdivp     pop    ebp     ret</pre>
--	---

