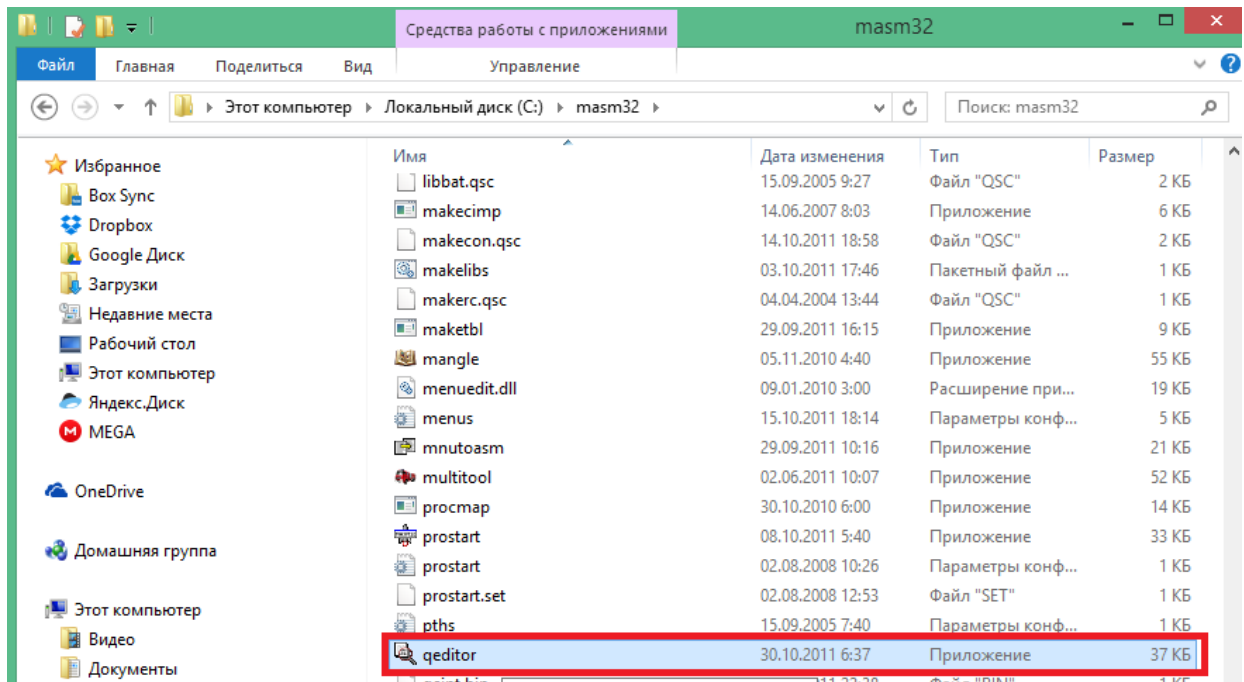
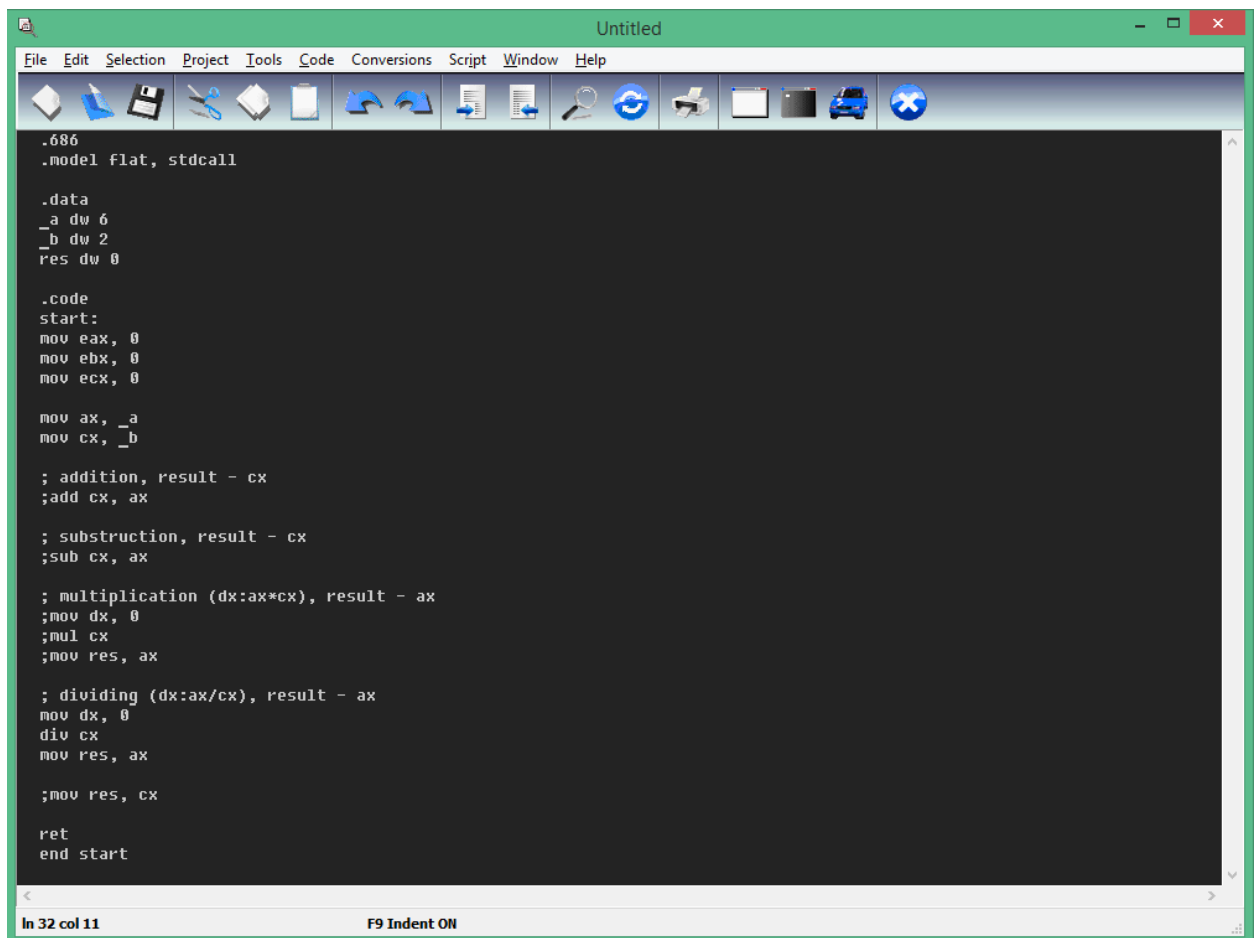


1. Запустить редактор MASM – qeditor



2. Набрать текст программы (в примере представлены все арифметические операции, раскомментирована и активна операция деления)



Код примера:

.686

.model flat, stdcall

.data

_a dw 6

_b dw 2

res dw 0

.code

start:

mov eax, 0

mov ebx, 0

mov ecx, 0

mov ax, _a

mov cx, _b

; addition, result - cx

;add cx, ax

; subtraction, result - cx

;sub cx, ax

; multiplication (dx:ax*cx), result - ax

;mov dx, 0

;mul cx

;mov res, ax

; dividing (dx:ax/cx), result - ax

mov dx, 0

div cx

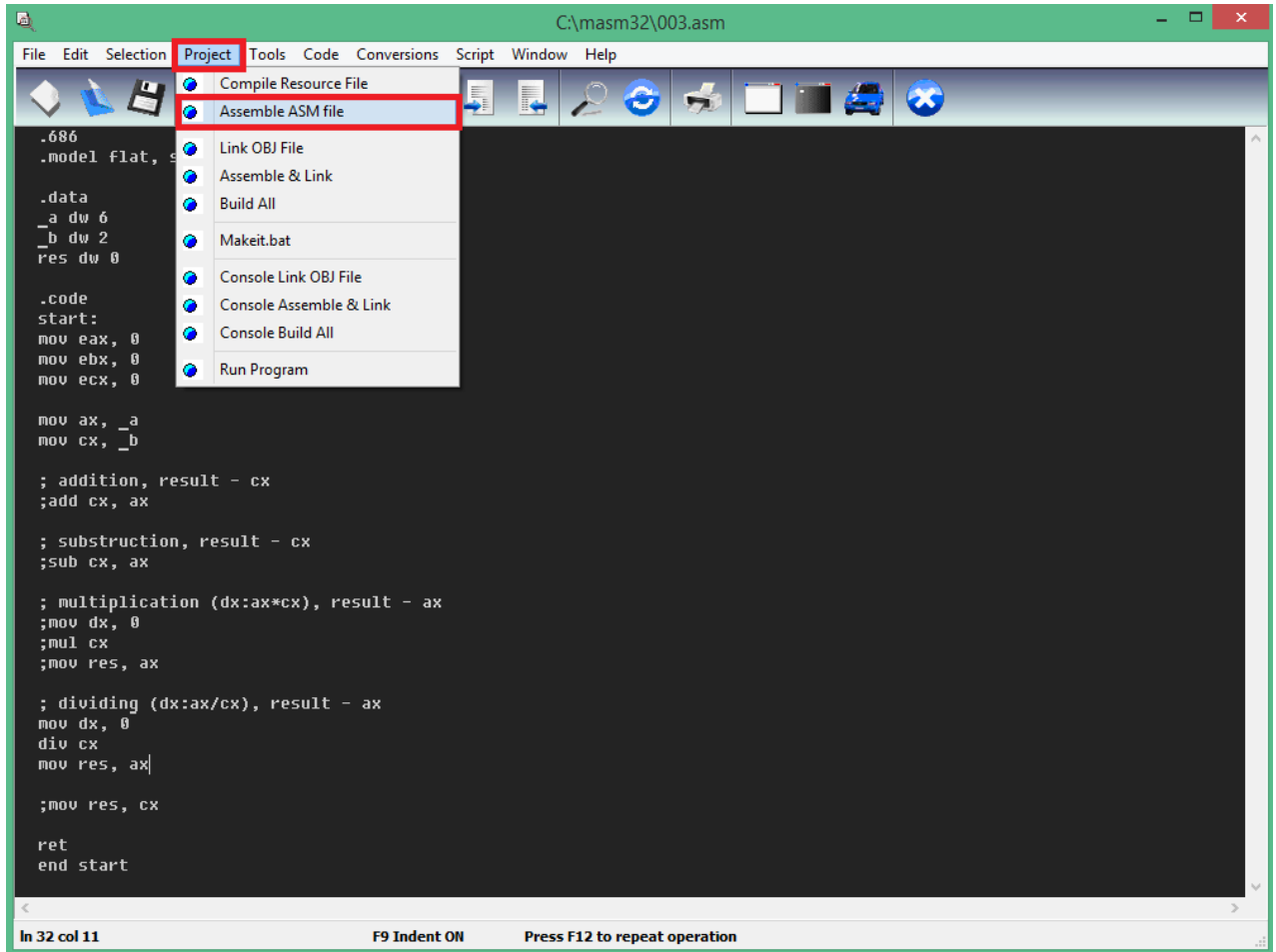
mov res, ax

;mov res, cx

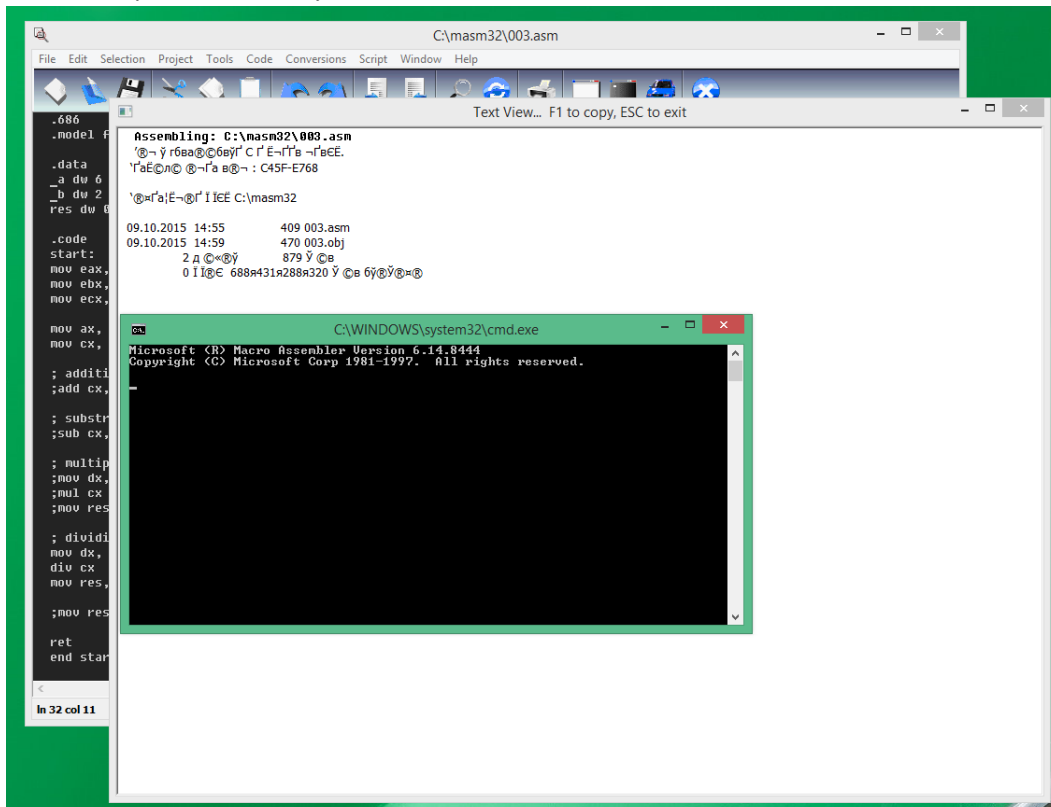
ret

end start

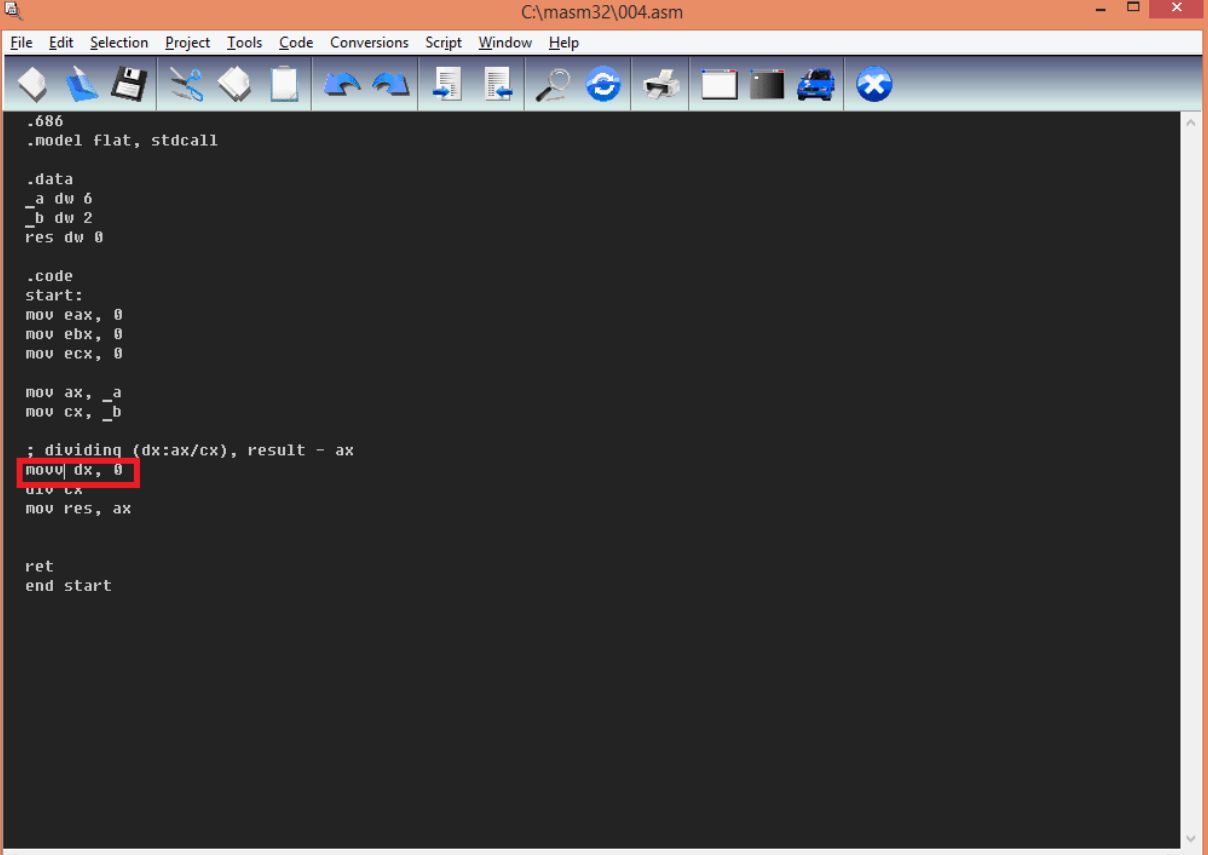
3. Сохранить файл: File-> Save as -> имя файла с расширением *.asm (расширение обязательно!)
4. Собрать асм-файл: Project-> Assemble ASM file



Если все правильно, получим:



Если допущена ошибка, то в текстовом отображении консоли (Text View) мы увидим строку, в которой допущена ошибка, и символы, которым она предшествует:



The screenshot shows the MASM32 IDE window titled "C:\masm32\004.asm". The menu bar includes File, Edit, Selection, Project, Tools, Code, Conversions, Script, Window, and Help. The toolbar contains icons for file operations and development tools. The assembly code is as follows:

```
.686
.model flat, stdcall

.data
_a dw 6
_b dw 2
res dw 0

.code
start:
mov eax, 0
mov ebx, 0
mov ecx, 0

mov ax, _a
mov cx, _b

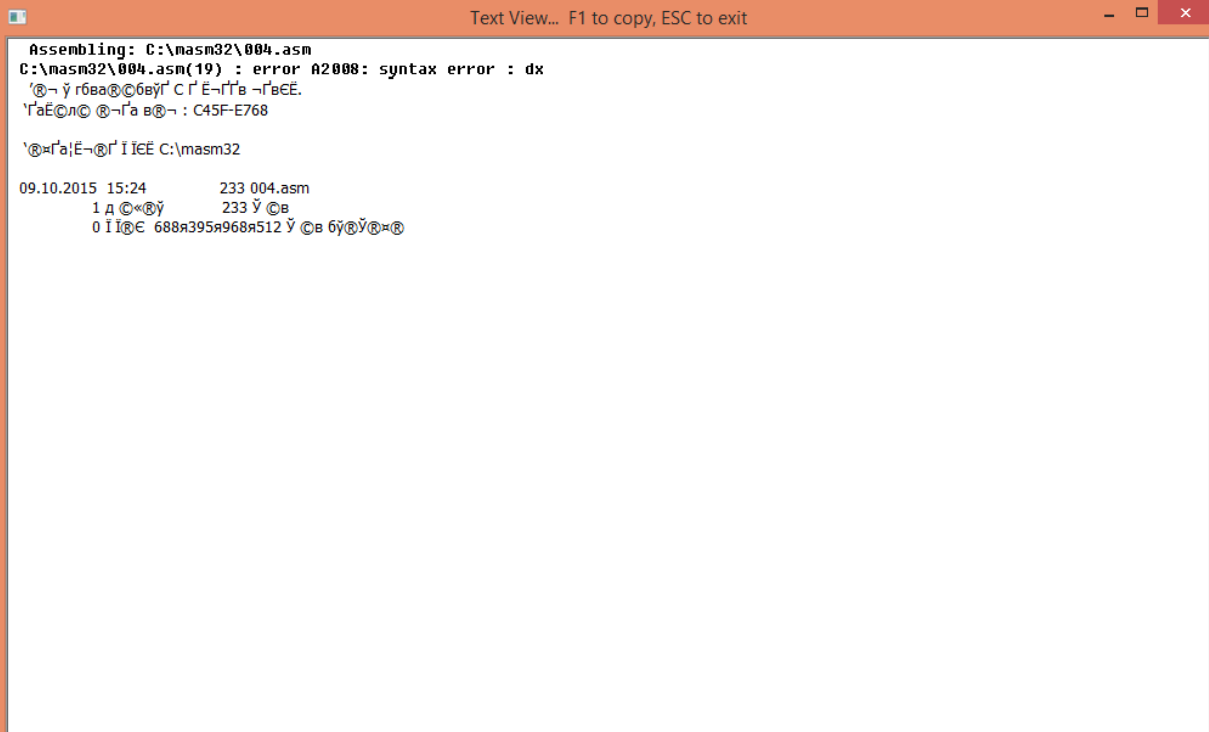
; dividing (dx:ax/cx), result - ax
mov dx, 0
div cx
mov res, ax

ret
end start
```

The instruction `mov dx, 0` on line 19 is highlighted with a red rectangle. The status bar at the bottom indicates "In 19 col 4", "F9 Indent ON", and "Press F12 to repeat operation".

Ошибка – `mov` с двумя «v».

При ассемблировании текстовый вид консоли принимает вид:



The screenshot shows the "Text View..." window titled "Text View... F1 to copy, ESC to exit". It displays the following error message:

```
Assembling: C:\masm32\004.asm
C:\masm32\004.asm(19) : error A2008: syntax error : dx
'0- Ÿ r6ba@06вŸГ' C Г' Е-ГГ'в -Г'вЕЕ.
Г'аЕ@л@ @-Г'а в@- : C45F-E768

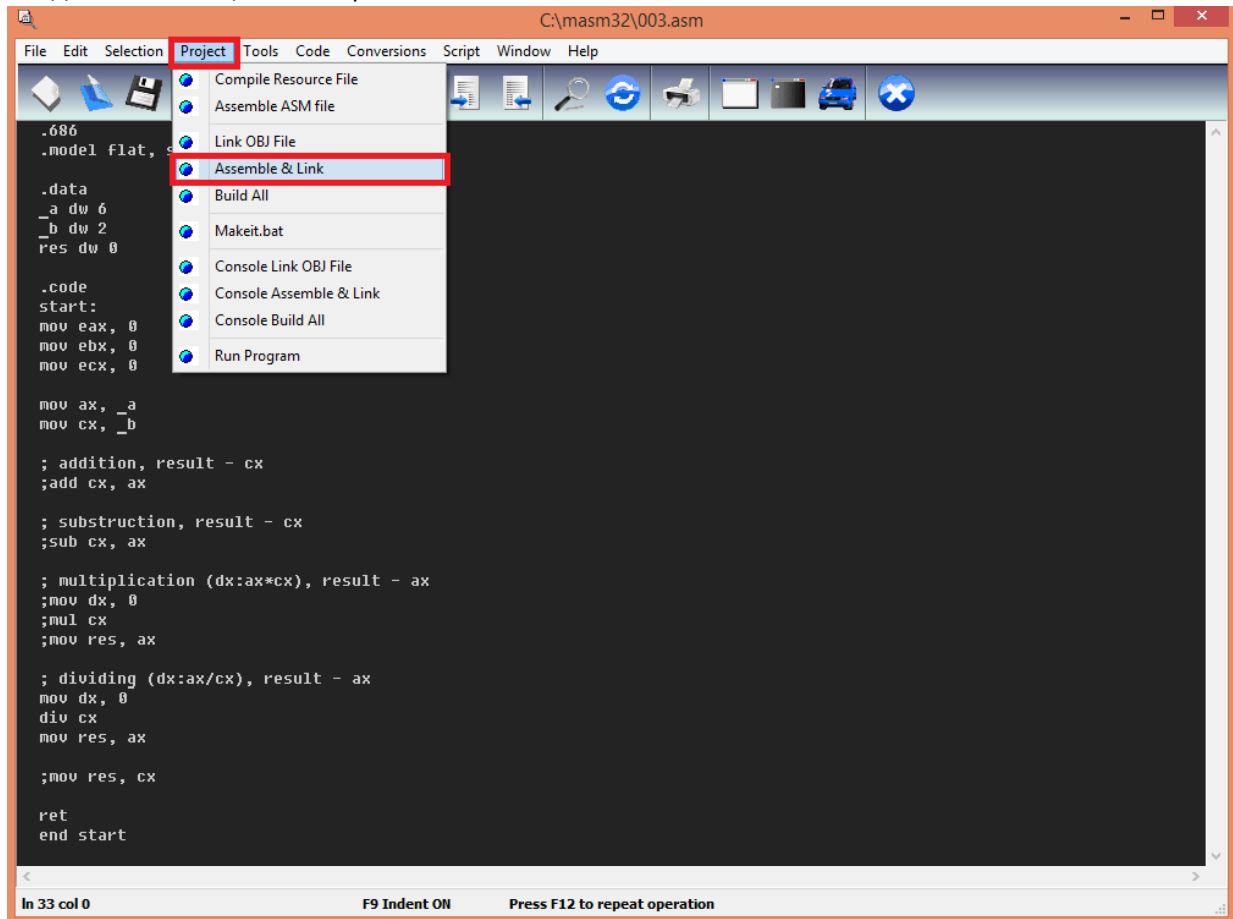
Г'аГ'аЕ-Г' Г'ГЕ C:\masm32

09.10.2015 15:24      233 004.asm
1 д @«@Ÿ      233 Ÿ @в
0 ГГ@С 688я395я968я512 Ÿ @в 6Ÿ@Ÿ@«@
```

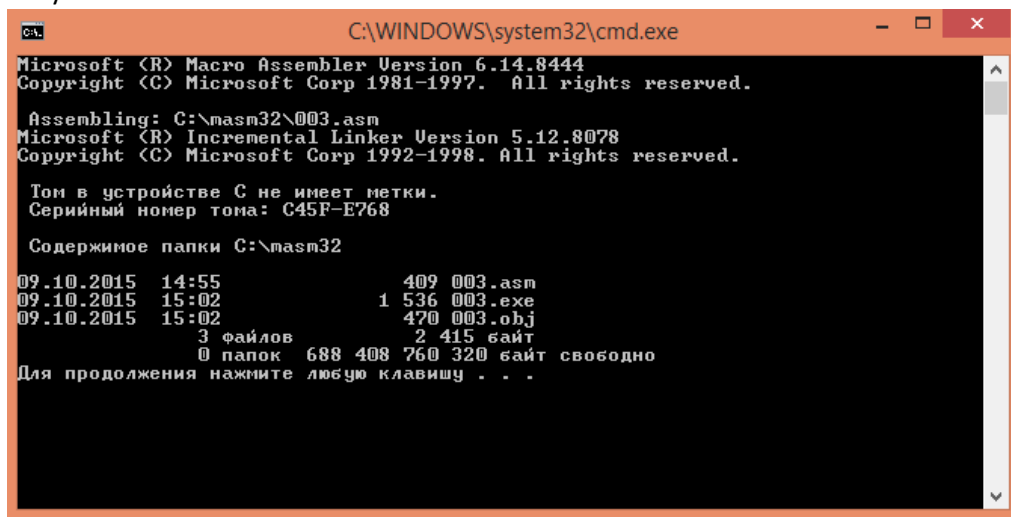
Это означает, что допущена синтаксическая ошибка где-то в строке 19 до `dx`.

Так как допущена ошибка, файл линковщика (*.obj) не создан.

5. Создаем линковщик и ехе-файл



Результат:



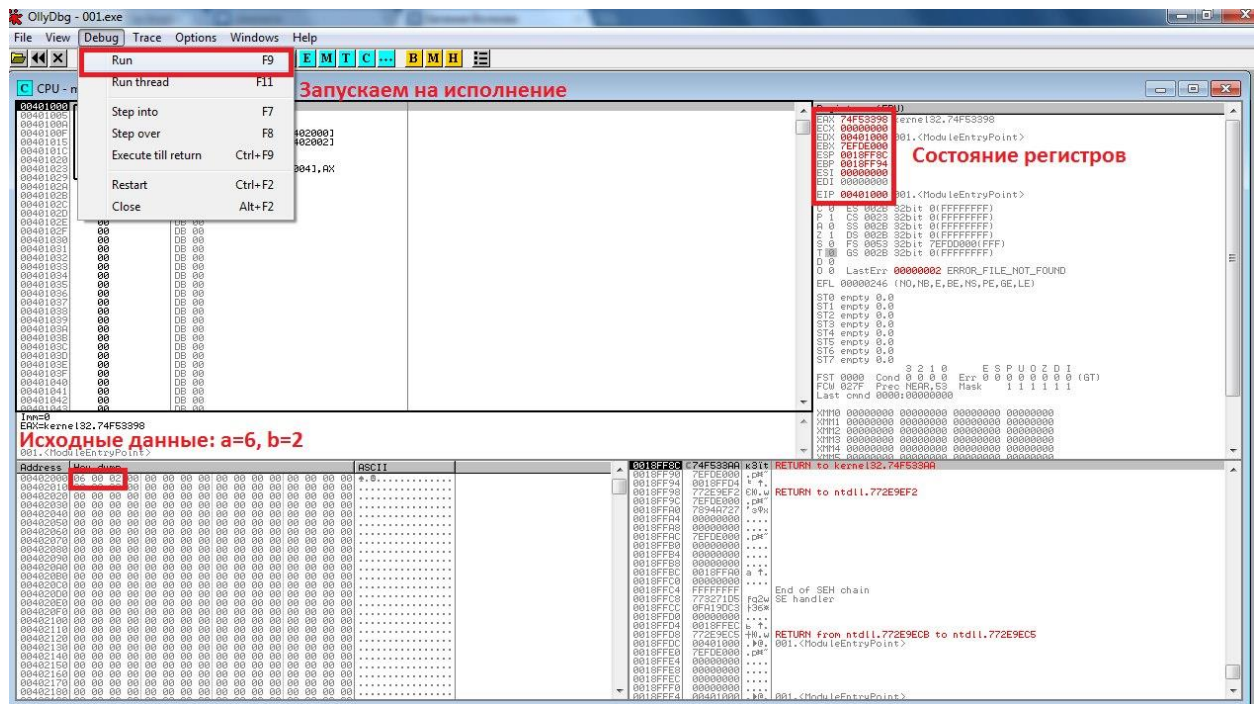
Внимание! Чтобы заново собрать файл, необходимо удалить obj и exe, и только после этого снова выполнить команду Project-> Assemble&Link.

6. Открываем Ollydbg (C:\Development\ollydbg – нужно разархивировать программу)

Внимание! Требуется запуск от имени администратора!

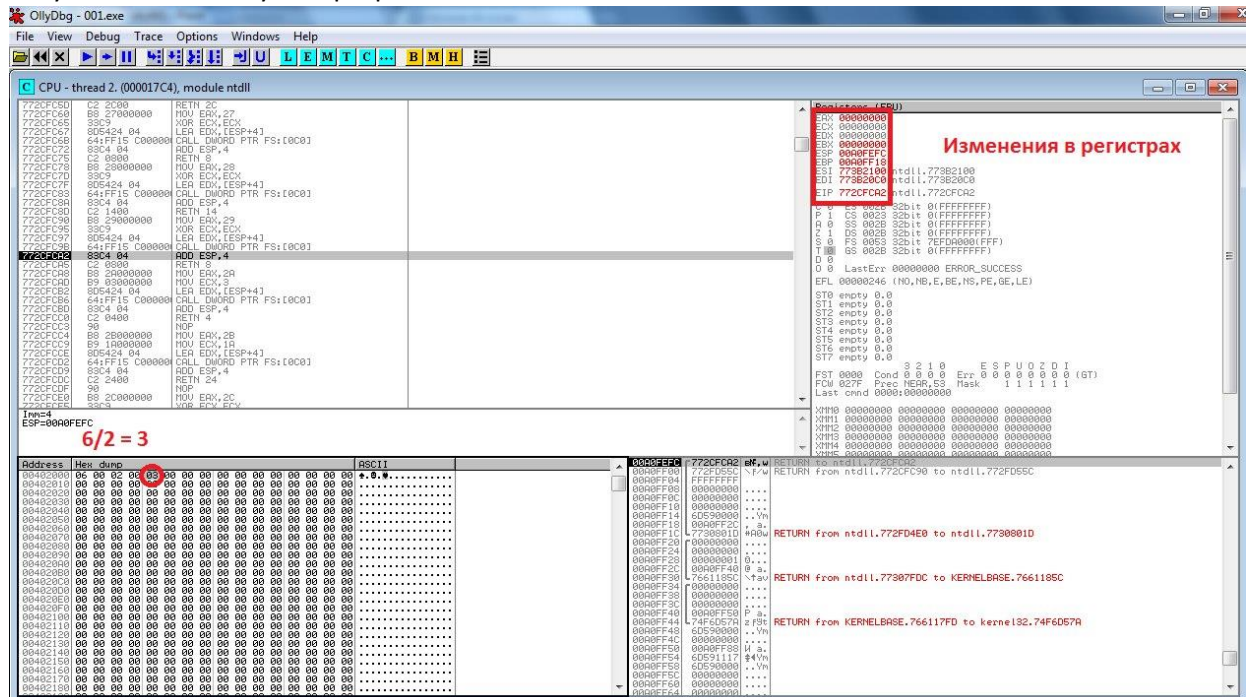
Открываем полученный в пункте 5 exe-файл: File-> Open

7. Далее запускаем файл на исполнение: Debug->Run

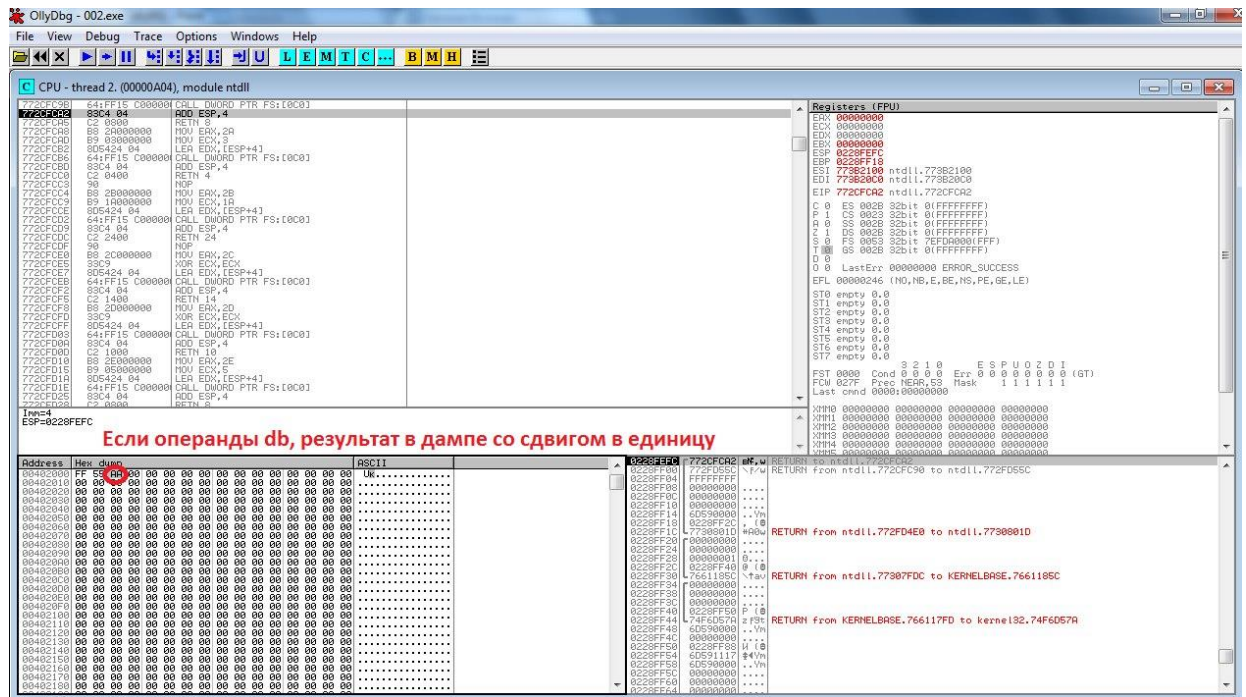


В нижней области размещен буфер (дамп) с данными, справа – состояние регистров процессора.

8. Результат после запуска программы:



Сдвиг между значениями в дампе определяется размером переменных, в случае с dw – под одно значение выделяется 2 байта. Если использовать db, то сдвиг будет равен 1 байту, значения будут идти подряд:



По сути, Ollydbg дизассемблирует exe-файл, поэтому код программы, который отображается в левой верхней части, может не совпадать на 100% с написанным на этапе 2 кодом.

- Добавим к созданной программе окно, в котором выводится результат.

Пример кода приведен в файле **shapka.asm**

Необходимо добавить к решению строки, выделенные красными в соответствующие части кода:

.386

.model flat, stdcall

option casemap :none

include \masm32\include\windows.inc

include \masm32\include\user32.inc

include \masm32\include\kernel32.inc

include \masm32\include\gdi32.inc

include \masm32\include\masm32.inc

includelib \masm32\lib\user32.lib

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\gdi32.lib

includelib \masm32\lib\masm32.lib

.data

_a dw

_b dw

_c dw

res dw 0

format db " Answer = %d",0

string db 30 DUP(?)

TextW db "Answer",0

.code

start:

```
invoke wsprintf,addr string,addr format,res  
invoke MessageBox,0,addr string,addr TextW,MB_OK
```

```
invoke ExitProcess,0
```

```
end start
```

Внимание! Для корректной работы программы ответ обязательно должен содержаться в ячейке памяти **res**.

10. Пересобираем проект (см. пункт 5) и запустить из qeditor (Project->Run Program). Убедиться, что программа работает, и в окне выводится верный результат.

По окончании выполнения работы необходимо написать отчет, который должен содержать:

- 1) ФИО, номер варианта, исходную функцию
- 2) Код программы
- 3) Скриншот выполненной программы в OllyDbg
- 4) Скриншот выполненной программы с ответом в окне

Сдать отчет и исходный код программы (файл .asm)