ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ МОЛДОВЫ

ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ, ИНФОРМАТИКИ И МИКРОЭЛЕКТРОНИКИ

КАФЕДРА АВТОМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЕТ

По лабораторной работе №3

по дисциплине «Архитектура компьютеров»

Тема: Основы языка Ассемблер.

Выполнила: ст. гр. TI-155 Зверкова К.

Проверил: Колесник В.

Кишинев 2017

**Цель лабораторной работы**

Работа рассматривает команды пересылки данных, команды арифметических операций над целыми двоичными числами и над двоично-десятичными числами.

**Пример 1.** **Вычислить арифметические выражения:**

INCLUDE Irvine32.inc

.data

mes1 byte "Enter the X:",0

mes2 byte "Enter the Y:",0

mes3 byte "Result:",0

vrx dword 0

vry dword 0

rez dd 0

.code

main PROC

mov edx,OFFSET mes1

call WriteString ; вввод mes1

call ReadDec ; ввод с клавиатуры

mov vrx,eax ; сохранение в переменную vrx

mov edx,OFFSET mes2

call WriteString ; вывод на экран mes2

call ReadDec ; ввод с клавиатуры 2-го значения

mov vry,eax ; сохранение в переменную vry

;проверяем условия

xor eax,eax

mov edx,0

mov eax,vry

mov bx,2

mul bx ; Y\*2

cmp vrx,eax ;сравнение X с 2Y

jb con1 ; переход к метке con1, если X<2Y

mov eax,vry ; 2Y-60

mov bx,2

mul bx

sub eax,60

mov rez,eax

jmp ex ; безусловный переход к ex

con1: mov eax,vrx ; X/8+32-Y

mov bx,8

div bx

add eax,32

sub eax,vry

mov rez,eax

ex: mov edx,OFFSET mes3

call WriteString ; вывод mes3

call WriteInt ; вывод результата со знаком

call Crlf ; с новой строки

exit

main ENDP

END main

**Листинг**

00000000 .data

00000000 45 6E 74 65 72 mes1 byte "Enter the X:",0

20 74 68 65 20

58 3A 00

0000000D 45 6E 74 65 72 mes2 byte "Enter the Y:",0

20 74 68 65 20

59 3A 00

0000001A 52 65 73 75 6C mes3 byte "Result:",0

74 3A 00

00000022 00000000 vrx dword 0

00000026 00000000 vry dword 0

0000002A 00000000 rez dd 0

00000000 .code

00000000 main PROC

00000000 BA 00000000 R mov edx,OFFSET mes1

00000005 E8 00000000 E call WriteString

0000000A E8 00000000 E call ReadDec

0000000F A3 00000022 R mov vrx,eax

00000014 BA 0000000D R mov edx,OFFSET mes2

00000019 E8 00000000 E call WriteString

0000001E E8 00000000 E call ReadDec

00000023 A3 00000026 R mov vry,eax

00000028 33 C0 xor eax,eax

0000002A BA 00000000 mov edx,0

0000002F A1 00000026 R mov eax,vry

00000034 66| BB 0002 mov bx,2

00000038 66| F7 E3 mul bx ; Y\*2

0000003B 39 05 00000022 R cmp vrx,eax

00000041 72 16 jb con1

00000043 A1 00000026 R mov eax,vry ; 2Y-60

00000048 66| BB 0002 mov bx,2

0000004C 66| F7 E3 mul bx

0000004F 83 E8 3C sub eax,60

00000052 A3 0000002A R mov rez,eax

00000057 EB 1A jmp ex

00000059 A1 00000022 R con1: mov eax,vrx ; X/8+32-Y

0000005E 66| BB 0008 mov bx,8

00000062 66| F7 F3 div bx

00000065 83 C0 20 add eax,32

00000068 2B 05 00000026 R sub eax,vry

0000006E A3 0000002A R mov rez,eax

00000073 BA 0000001A R ex: mov edx,OFFSET mes3

00000078 E8 00000000 E call WriteString

0000007D E8 00000000 E call WriteInt

00000082 E8 00000000 E call Crlf

exit

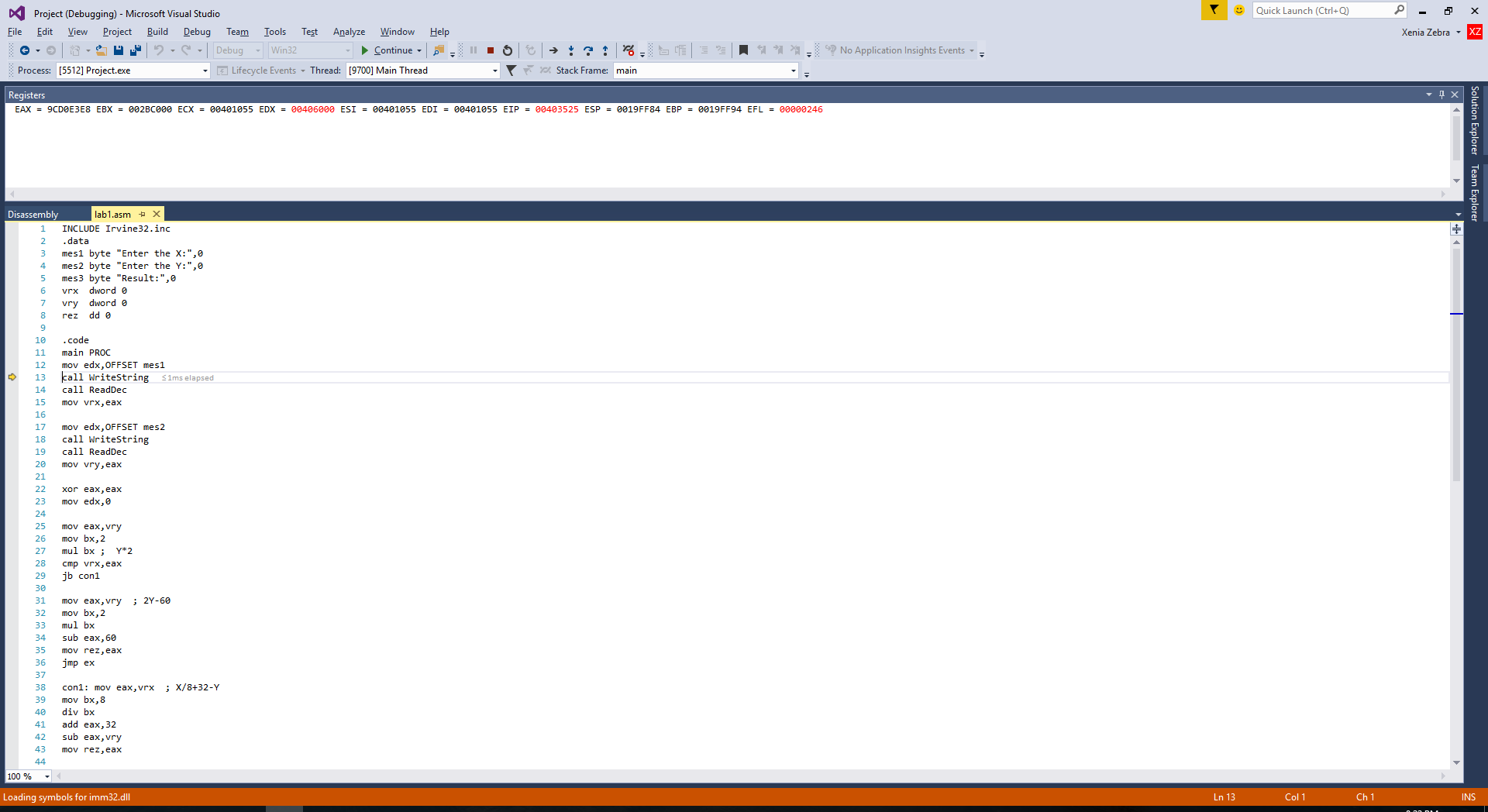
00000087 6A 00 \* push +000000000h

00000089 E8 00000000 E \* call ExitProcess

0000008E main ENDP

END main

**Дебаг**



**Пример 2. Есть три целых числа (абс) определенные как три последовательных слова. Необходимо произвести циклический сдвиг этих чисел влево (бса).**

INCLUDE Irvine32.inc

.DATA

vect WORD 100h,200h,300h

.CODE

main PROC

mov dx,vect ; запись в dx первого слова переменной vect

mov ax,vect+2; запись 2-го слова

mov vect,ax ; по адресу первого

mov ax,vect+4; запись 3-го слова

mov vect+2,ax; по адресу второго

mov vect+4,dx; запись 1-го слова по адресу 3-го

exit

main ENDP

END main

**Листинг**

00000000 .DATA

00000000 0100 0200 0300 vect WORD 100h,200h,300h

00000000 .CODE

00000000 main PROC

00000000 66| 8B 15 mov dx,vect

00000000 R

00000007 66| A1 mov ax,vect+2

00000002 R

0000000D 66| A3 mov vect,ax

00000000 R

00000013 66| A1 mov ax,vect+4

00000004 R

00000019 66| A3 mov vect+2,ax

00000002 R

0000001F 66| 89 15 mov vect+4,dx

00000004 R

exit

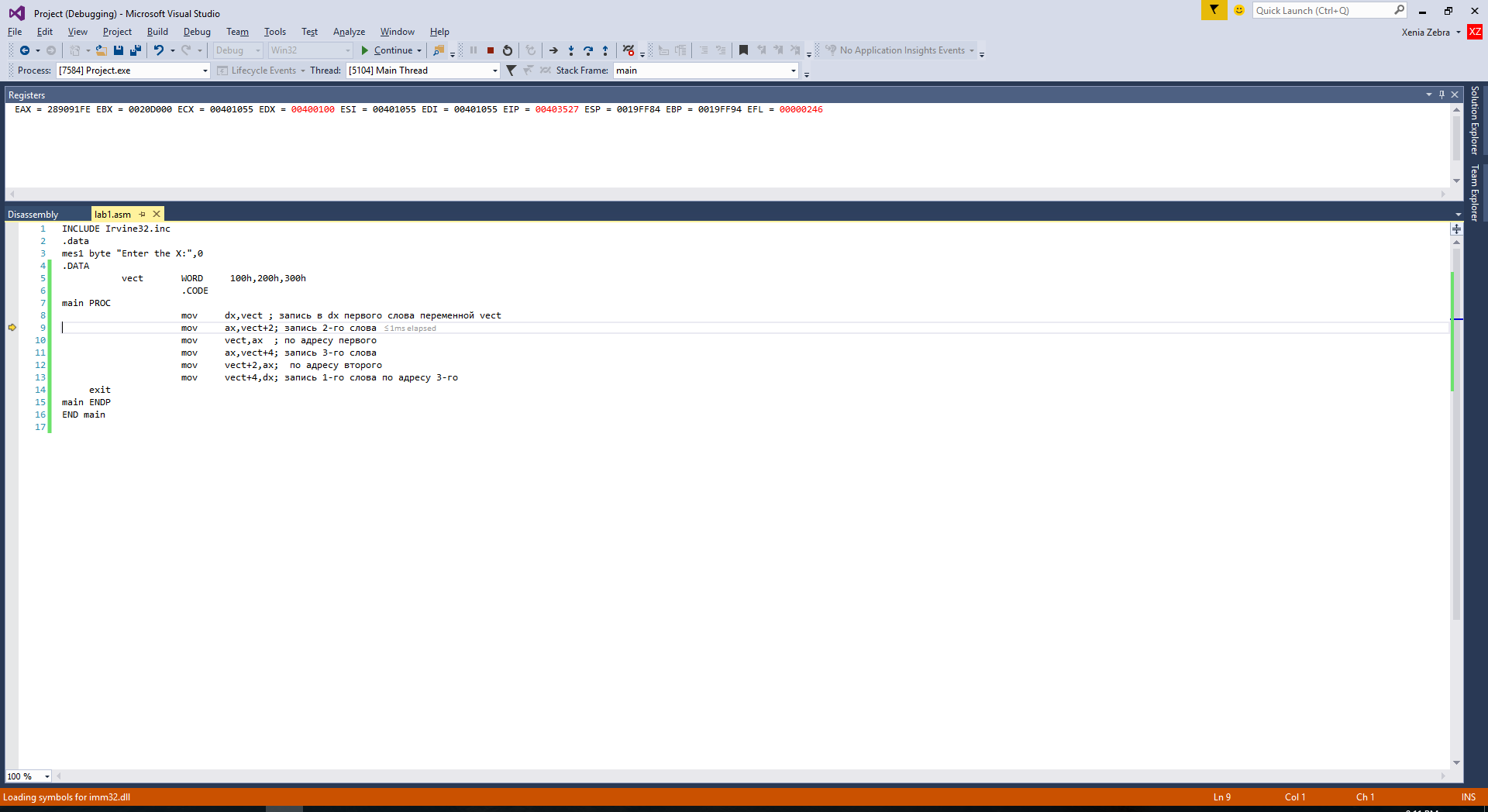
00000026 6A 00 \* push +000000000h

00000028 E8 00000000 E \* call ExitProcess

0000002D main ENDP

END main

**Дебаг**



**Пример 3. Есть три слова в памяти. Необходимо записать значение 0ааh в старшие байты этих слов.**

INCLUDE Irvine32.inc

.DATA

val EQU 0aah

vect DW 111h,222h,333h

.CODE

main PROC

mov al,val ; записывается значение 0aah

; в регистр al

mov BYTE PTR vect[1],al ; запись в

mov BYTE PTR vect[3],al ; старшие

mov BYTE PTR vect[5],al ; байты

exit

main ENDP

END main

**Листинг**

00000000 .DATA

= 000000AA val EQU 0aah

00000000 0111 0222 0333 vect DW 111h,222h,333h

00000000 .CODE

00000000 main PROC

00000000 B0 AA mov al,val

00000002 A2 00000001 R mov BYTE PTR vect[1],al

00000007 A2 00000003 R mov BYTE PTR vect[3],al

0000000C A2 00000005 R mov BYTE PTR vect[5],al

exit

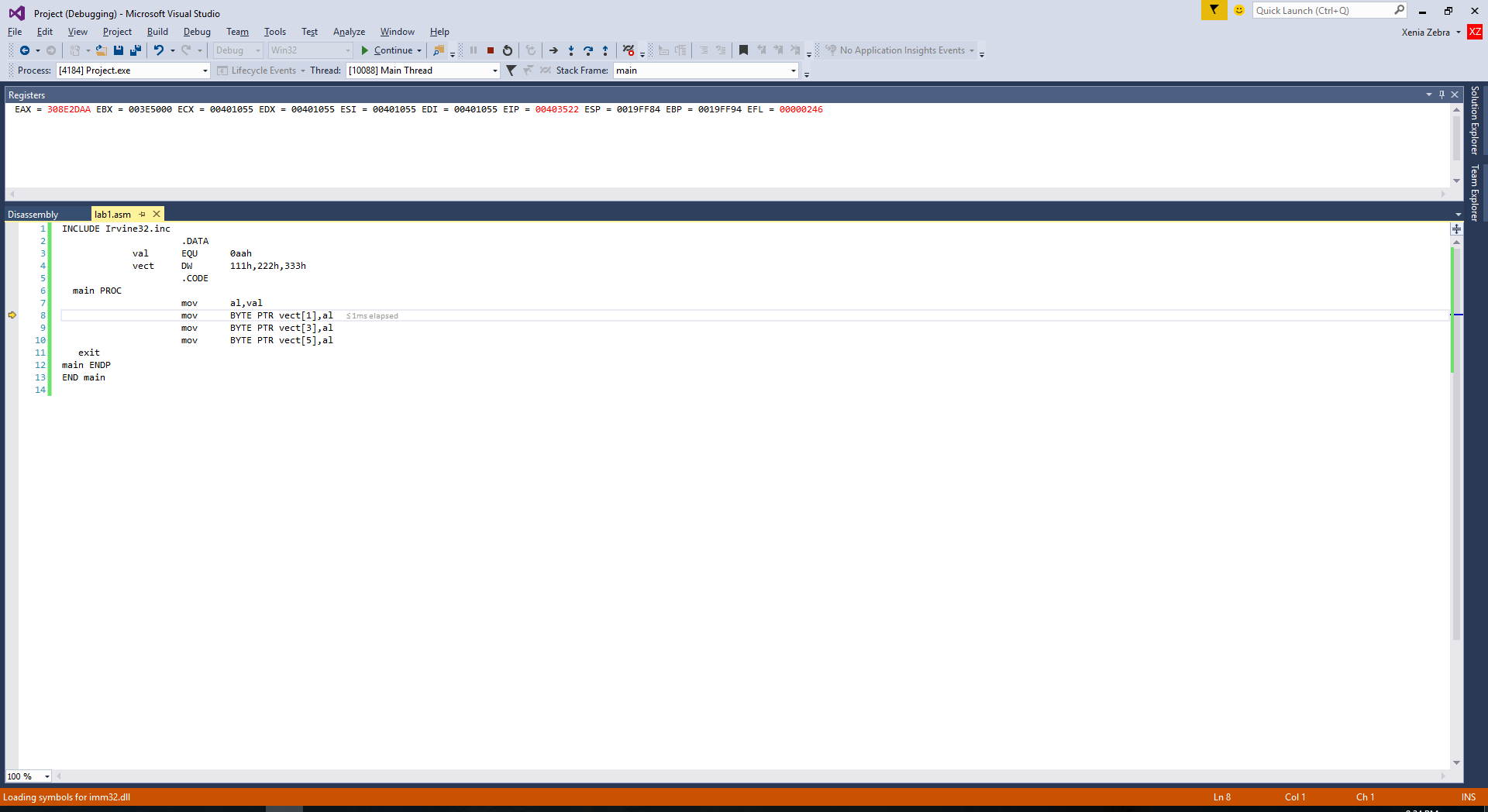
00000011 6A 00 \* push +000000000h

00000013 E8 00000000 E \* call ExitProcess

00000018 main ENDP

END main

**Дебаг**



**Пример 4. Дано 2 числа по 4 цифры в упакованном BCD формате (одно число два байта). Необходимо разработать программу, исполняющую суммирование и вычитания этих чисел.**

INCLUDE Irvine32.inc

.DATA

bcd1 BYTE 34h,18h; 1834 в упакованном BCD формате

bcd2 BYTE 89h,27h; 2789 в упакованном BCD формате

sum BYTE 2 DUP(?)

dif BYTE 2 DUP(?)

.CODE

main PROC

mov al,bcd1

add al,bcd2

daa

mov sum,al

mov al,bcd1+1

adc al,bcd2+1

daa

mov sum+1,al

mov al,bcd1

sub al,bcd2

das

mov dif,al

mov al,bcd1+1

sbb al,bcd2+1

das

mov sum+1,al

exit

main ENDP

END main

**Листинг**

00000000 .DATA

00000000 34 18 bcd1 BYTE 34h,18h

00000002 89 27 bcd2 BYTE 89h,27h

00000004 00000002 [ sum BYTE 2 DUP(?)

00

]

00000006 00000002 [ dif BYTE 2 DUP(?)

00

]

00000000 .CODE

00000000 main PROC

00000000 A0 00000000 R mov al,bcd1

00000005 02 05 00000002 R add al,bcd2

0000000B 27 daa

0000000C A2 00000004 R mov sum,al

00000011 A0 00000001 R mov al,bcd1+1

00000016 12 05 00000003 R adc al,bcd2+1

0000001C 27 daa

0000001D A2 00000005 R mov sum+1,al

00000022 A0 00000000 R mov al,bcd1

00000027 2A 05 00000002 R sub al,bcd2

0000002D 2F das

0000002E A2 00000006 R mov dif,al

00000033 A0 00000001 R mov al,bcd1+1

00000038 1A 05 00000003 R sbb al,bcd2+1

0000003E 2F das

0000003F A2 00000005 R mov sum+1,al

exit

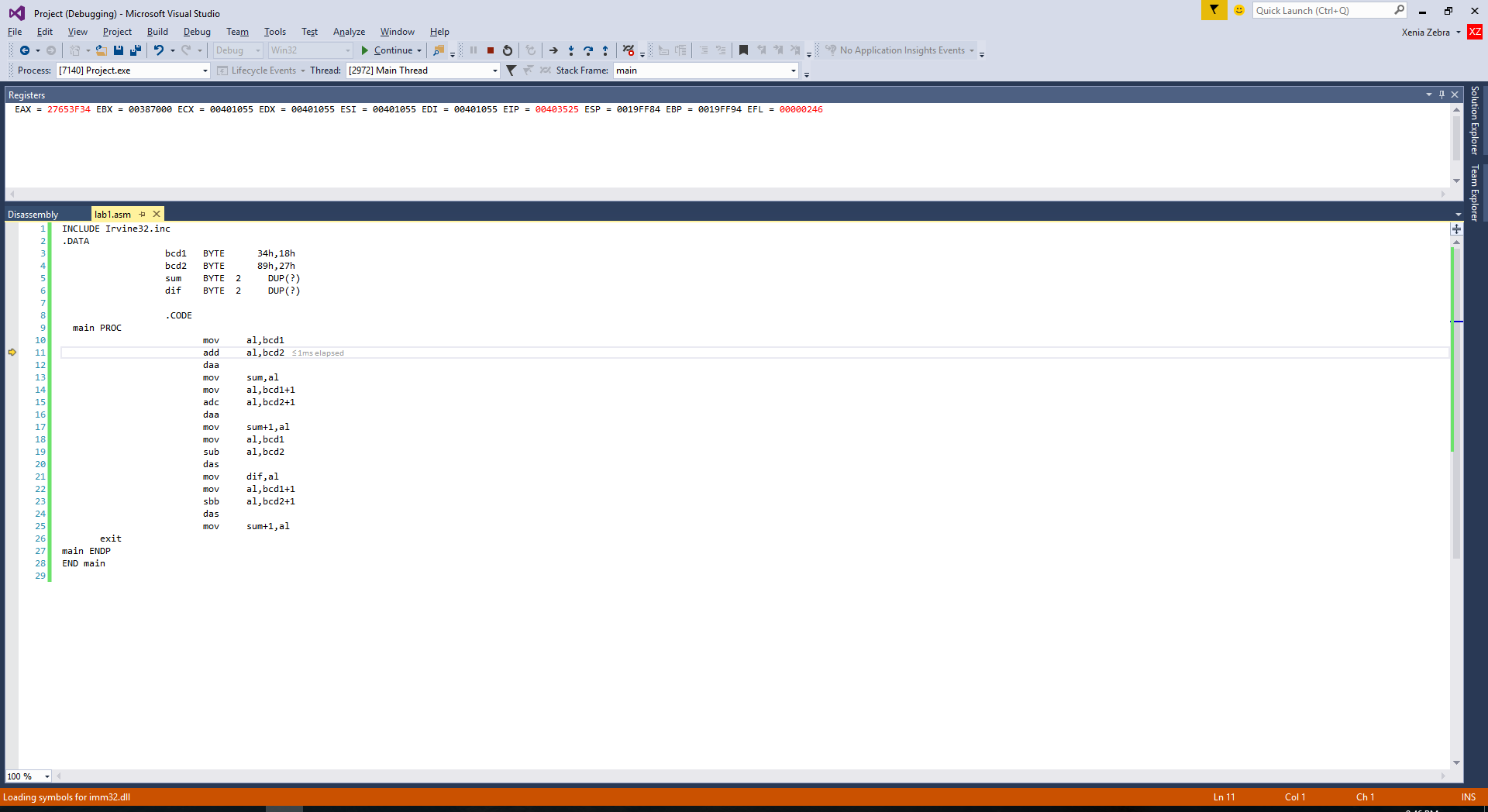
00000044 6A 00 \* push +000000000h

00000046 E8 00000000 E \* call ExitProcess

0000004B main ENDP

END main

**Дебаг**



**Пример 5. Разработать программу умножения двух положительных чисел в неупакованном BCD формате.**

INCLUDE Irvine32.inc

.DATA

npk1 DB 09h ; 9 в неупакованном BCD формате

npk2 DB 03h,02h; 23 в неупакованном BCD формате

rez DB 3 DUP(?)

.CODE

main PROC

mov dl,npk1

mov al,npk2

mul dl

aam

mov dh,ah

mov rez,al

mov al,npk2+1

mul dl

aam

add al,dh

aaa

mov rez+1,al

mov rez+2,ah

exit

main ENDP

END main

**Листинг**

00000000 .DATA

00000000 09 npk1 DB 09h

00000001 03 02 npk2 DB 03h,02h

00000003 00000003 [ rez DB 3 DUP(?)

00

]

00000000 .CODE

00000000 main PROC

00000000 8A 15 00000000 R mov dl,npk1

00000006 A0 00000001 R mov al,npk2

0000000B F6 E2 mul dl

0000000D D4 0A aam

0000000F 8A F4 mov dh,ah

00000011 A2 00000003 R mov rez,al

00000016 A0 00000002 R mov al,npk2+1

0000001B F6 E2 mul dl

0000001D D4 0A aam

0000001F 02 C6 add al,dh

00000021 37 aaa

00000022 A2 00000004 R mov rez+1,al

00000027 88 25 00000005 R mov rez+2,ah

exit

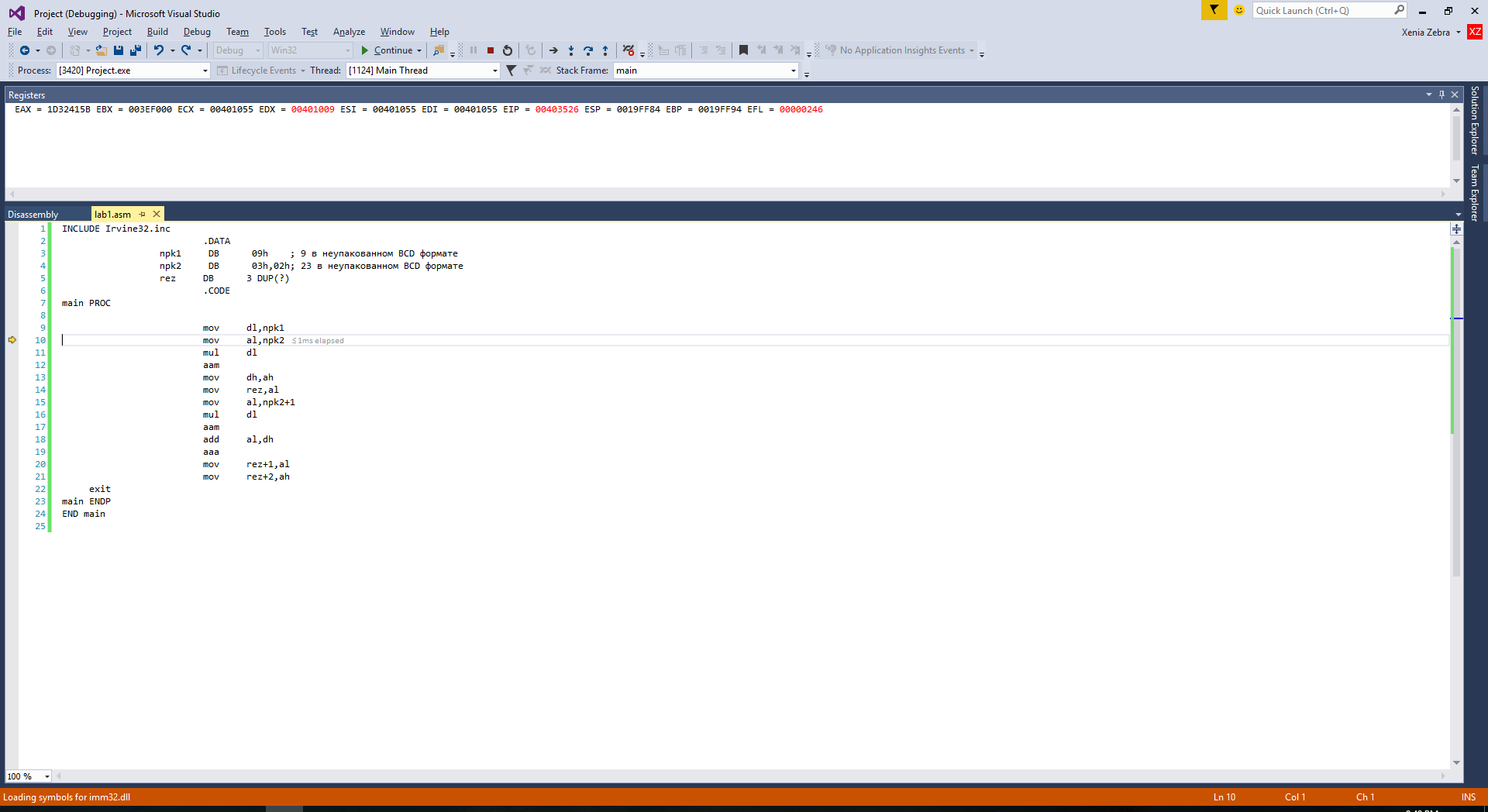
0000002D 6A 00 \* push +000000000h

0000002F E8 00000000 E \* call ExitProcess

00000034 main ENDP

END main

**Дебаг**



**В соответствии с вариантом индивидуального задания разработать 2 варианта программ:**

1. **ввод данных с клавиатуры и вывод результата на экран.**
2. **с генерацией входных данных, используя процедуры Randomize, Random32, RandomRange.**

**1)**

INCLUDE Irvine32.inc

.data

mes1 byte "Enter the X:",0

mes2 byte "Enter the Y:",0

mes3 byte "Result:",0

vrx dword 0

vry dword 0

vrt dword 0

rez dd 0

.code

main PROC

mov edx,OFFSET mes1

call WriteString ; вввод mes1

call ReadDec ; ввод с клавиатуры

mov vrx,eax ; сохранение в переменную vrx

mov edx,OFFSET mes2

call WriteString ; вывод на экран mes2

call ReadDec ; ввод с клавиатуры 2-го значения

mov vry,eax ; сохранение в переменную vry

;проверяем условия

xor eax,eax

mov edx,0

mov eax,vry

mov bx,2

mul bx ; Y\*2

cmp vrx,eax ;сравнение X с 2Y

ja con1 ; переход к метке con1, если X>2Y

mov eax,vry ; 2Y+X/4

mov bx,2

mul bx

mov vrt, eax ; 2Y

mov eax, vrx

mov bx, 4

div bx

add eax, vrt

mov rez,eax

jmp ex ; безусловный переход к ex

con1: mov eax,vrx ; 2(X-Y)+81

sub eax,vry

mov bx,2

mul bx

add eax,81

mov rez,eax

ex: mov edx,OFFSET mes3

call WriteString ; вывод mes3

call WriteInt ; вывод результата со знаком

call Crlf ; с новой строки

exit

main ENDP

END main

**Листинг**

00000000 .data

00000000 45 6E 74 65 72 mes1 byte "Enter the X:",0

20 74 68 65 20

58 3A 00

0000000D 45 6E 74 65 72 mes2 byte "Enter the Y:",0

20 74 68 65 20

59 3A 00

0000001A 52 65 73 75 6C mes3 byte "Result:",0

74 3A 00

00000022 00000000 vrx dword 0

00000026 00000000 vry dword 0

0000002A 00000000 vrt dword 0

0000002E 00000000 rez dd 0

00000000 .code

00000000 main PROC

00000000 BA 00000000 R mov edx,OFFSET mes1

00000005 E8 00000000 E call WriteString

0000000A E8 00000000 E call ReadDec

0000000F A3 00000022 R mov vrx,eax

00000014 BA 0000000D R mov edx,OFFSET mes2

00000019 E8 00000000 E call WriteString

0000001E E8 00000000 E call ReadDec

00000023 A3 00000026 R mov vry,eax

00000028 33 C0 xor eax,eax

0000002A BA 00000000 mov edx,0

0000002F A1 00000026 R mov eax,vry

00000034 66| BB 0002 mov bx,2

00000038 66| F7 E3 mul bx ; Y\*2

0000003B 39 05 00000022 R cmp vrx,eax

00000041 77 2A ja con1

00000043 A1 00000026 R mov eax,vry

00000048 66| BB 0002 mov bx,2

0000004C 66| F7 E3 mul bx

0000004F A3 0000002A R mov vrt, eax

00000054 A1 00000022 R mov eax, vrx

00000059 66| BB 0004 mov bx, 4

0000005D 66| F7 F3 div bx

00000060 03 05 0000002A R add eax, vrt

00000066 A3 0000002E R mov rez,eax

0000006B EB 1A jmp ex

0000006D A1 00000022 R con1: mov eax,vrx

00000072 2B 05 00000026 R sub eax,vry

00000078 66| BB 0002 mov bx,2

0000007C 66| F7 E3 mul bx

0000007F 83 C0 51 add eax,81

00000082 A3 0000002E R mov rez,eax

00000087 BA 0000001A R ex: mov edx,OFFSET mes3

0000008C E8 00000000 E call WriteString

00000091 E8 00000000 E call WriteInt

00000096 E8 00000000 E call Crlf

exit

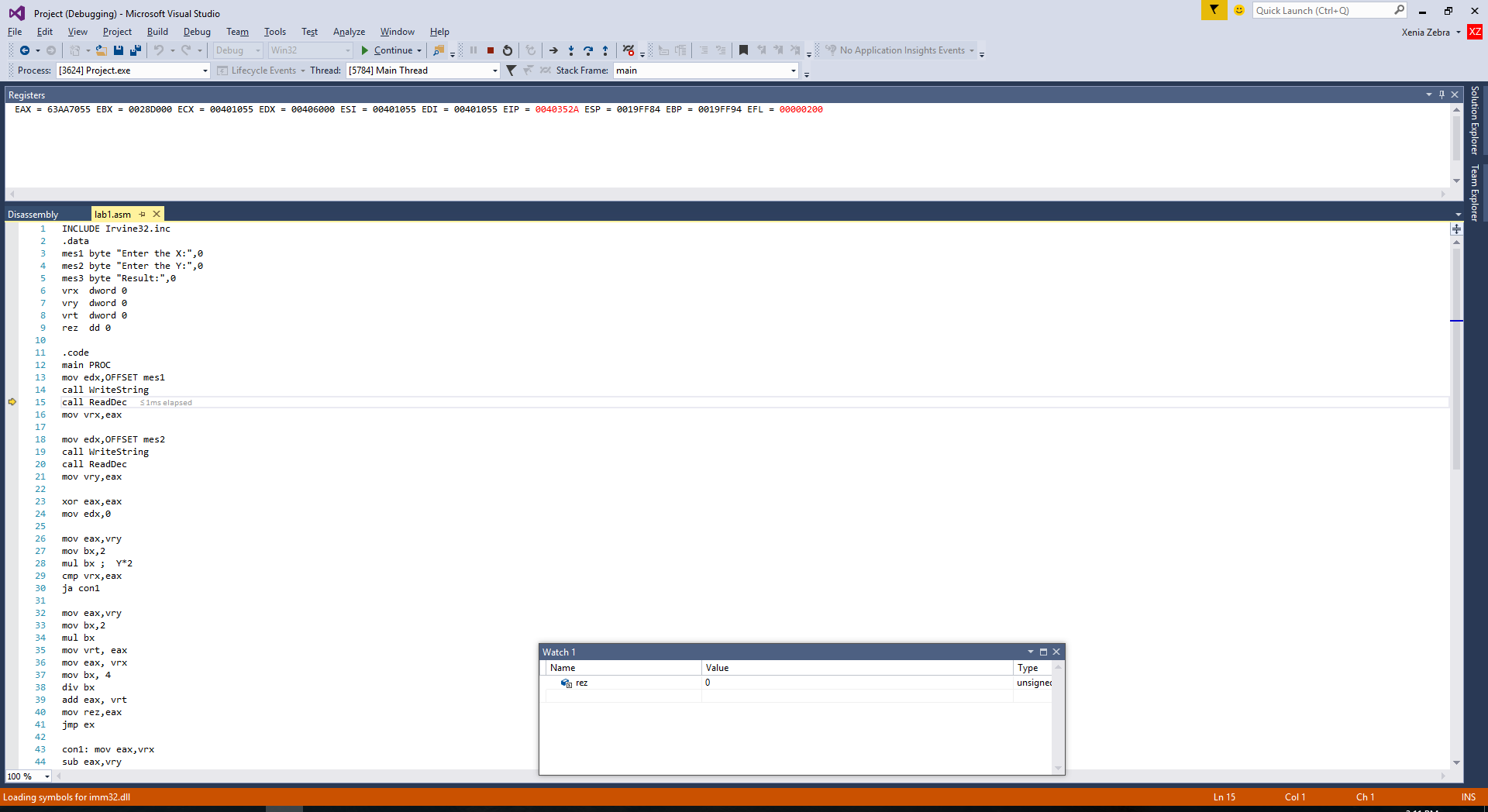
0000009B 6A 00 \* push +000000000h

0000009D E8 00000000 E \* call ExitProcess

000000A2 main ENDP

END main

**Дебаг**



**2)**

INCLUDE Irvine32.inc

.data

mes1 byte "Enter the X:",0

mes2 byte "Enter the Y:",0

mes3 byte "Result:",0

vrx dword 0

vry dword 0

vrt dword 0

rez dd 0

.code

main PROC

mov edx,OFFSET mes1

call WriteString

mov eax, 0

call Randomize

call Random32

mov vrx,eax

call WriteDec

call Crlf

mov edx,OFFSET mes2

call WriteString

mov eax,5000

call RandomRange

mov vry,eax

call WriteDec

call Crlf

xor eax,eax

mov edx,0

mov eax,vry

mov bx,2

mul bx ; Y\*2

cmp vrx,eax

ja con1

mov eax,vry

mov bx,2

mul bx

mov vrt, eax

mov eax, vrx

mov bx, 4

div bx

add eax, vrt

mov rez,eax

jmp ex

con1: mov eax,vrx

sub eax,vry

mov ebx,2

mul ebx

add eax,81

mov rez,eax

ex: mov edx,OFFSET mes3

call WriteString

call WriteInt

call Crlf

exit

main ENDP

END main

**Листинг**

00000000 .data

00000000 45 6E 74 65 72 mes1 byte "Enter the X:",0

20 74 68 65 20

58 3A 00

0000000D 45 6E 74 65 72 mes2 byte "Enter the Y:",0

20 74 68 65 20

59 3A 00

0000001A 52 65 73 75 6C mes3 byte "Result:",0

74 3A 00

00000022 00000000 vrx dword 0

00000026 00000000 vry dword 0

0000002A 00000000 vrt dword 0

0000002E 00000000 rez dd 0

00000000 .code

00000000 main PROC

00000000 BA 00000000 R mov edx,OFFSET mes1

00000005 E8 00000000 E call WriteString

0000000A B8 00000000 mov eax, 0

0000000F E8 00000000 E call Randomize

00000014 E8 00000000 E call Random32

00000019 A3 00000022 R mov vrx,eax

0000001E E8 00000000 E call WriteDec

00000023 E8 00000000 E call Crlf

00000028 BA 0000000D R mov edx,OFFSET mes2

0000002D E8 00000000 E call WriteString

00000032 B8 00001388 mov eax,5000

00000037 E8 00000000 E call RandomRange

0000003C A3 00000026 R mov vry,eax

00000041 E8 00000000 E call WriteDec

00000046 E8 00000000 E call Crlf

0000004B 33 C0 xor eax,eax

0000004D BA 00000000 mov edx,0

00000052 A1 00000026 R mov eax,vry

00000057 66| BB 0002 mov bx,2

0000005B 66| F7 E3 mul bx ; Y\*2

0000005E 39 05 00000022 R cmp vrx,eax

00000064 77 2A ja con1

00000066 A1 00000026 R mov eax,vry

0000006B 66| BB 0002 mov bx,2

0000006F 66| F7 E3 mul bx

00000072 A3 0000002A R mov vrt, eax

00000077 A1 00000022 R mov eax, vrx

0000007C 66| BB 0004 mov bx, 4

00000080 66| F7 F3 div bx

00000083 03 05 0000002A R add eax, vrt

00000089 A3 0000002E R mov rez,eax

0000008E EB 1A jmp ex

00000090 A1 00000022 R con1: mov eax,vrx

00000095 2B 05 00000026 R sub eax,vry

0000009B BB 00000002 mov ebx,2

000000A0 F7 E3 mul ebx

000000A2 83 C0 51 add eax,81

000000A5 A3 0000002E R mov rez,eax

000000AA BA 0000001A R ex: mov edx,OFFSET mes3

000000AF E8 00000000 E call WriteString

000000B4 E8 00000000 E call WriteInt

000000B9 E8 00000000 E call Crlf

exit

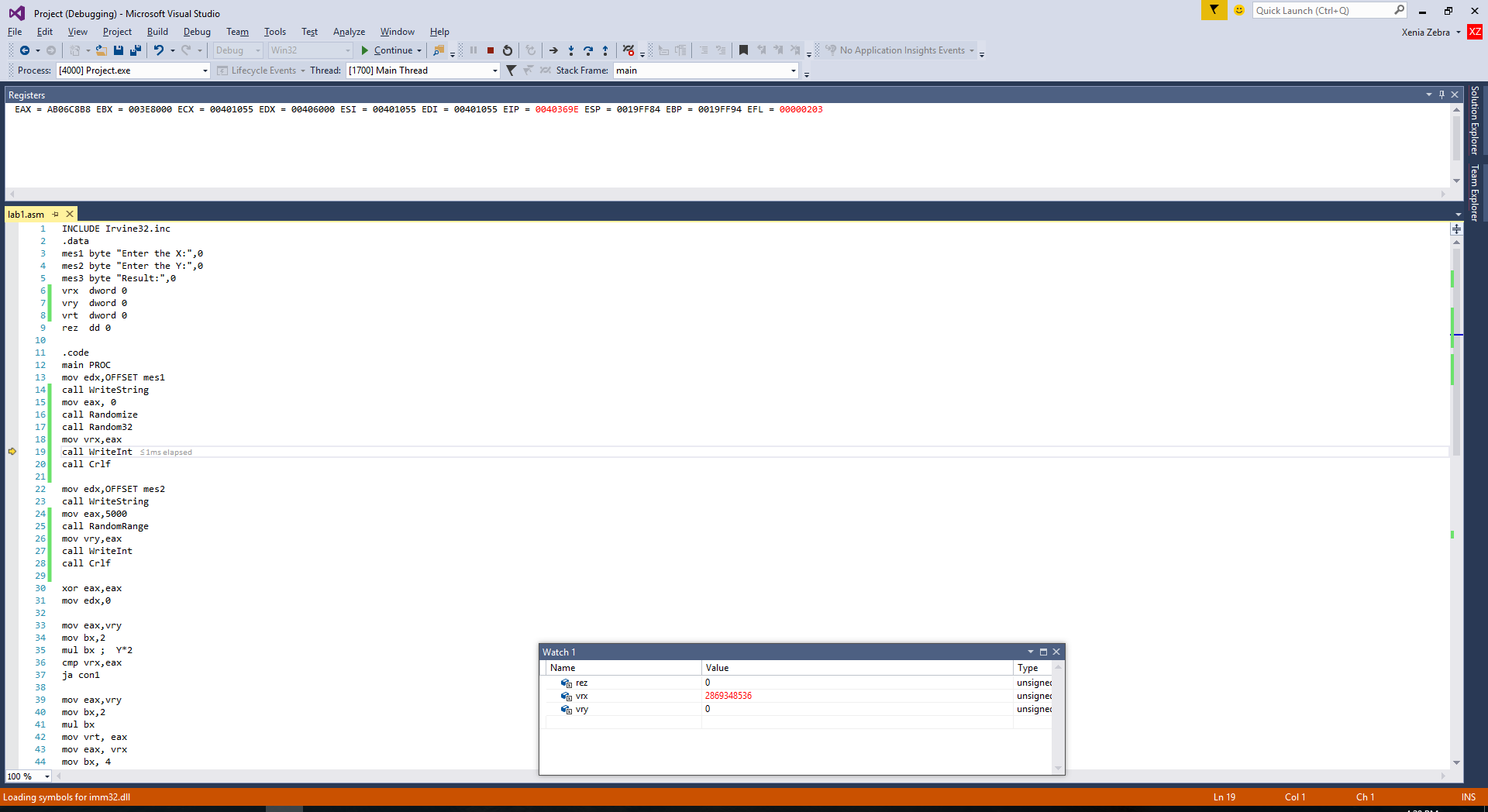
000000BE 6A 00 \* push +000000000h

000000C0 E8 00000000 E \* call ExitProcess

000000C5 main ENDP

END main

**Дебаг**



**Вывод:**

В ходе данной лабораторной работы были рассматрены команды пересылки данных, команды арифметических операций над целыми двоичными числами и над двоично-десятичными числами.На основе знаний, которые были получены были разработаны 2 варианта программ,вычисляющих арифметическое выражение: с вводом данных с клавиатуры и выводом результата на экран, с генерацией входных данных, используя процедуры Randomize, Random32, RandomRange.