

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Организация ЭВМ и систем»**  
**ТЕМА: Изучение режимов адресации и формирования**  
**исполнительного адреса.**

Студентка гр. 1303

Хабибуллина А.М.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Научиться работать с режимами адресации на языке программирования Ассемблер.

### **Задание.**

Лабораторная работа 2 предназначена для изучения режимов адресации, использует готовую программу lr2\_comp.asm на Ассемблере, которая в автоматическом режиме выполняться не должна, так как не имеет самостоятельного функционального назначения, а только тестирует режимы адресации. Поэтому ее выполнение должно производиться под управлением отладчика в пошаговом режиме. В программу введен ряд ошибок, которые необходимо объяснить в отчете по работе, а соответствующие команды закомментировать для прохождения трансляции. Необходимо составить протокол выполнения программы в пошаговом режиме отладчика по типу таблицы 1 предыдущей лабораторной работы и подписать его у преподавателя. На защите студенты должны уметь объяснить результат выполнения каждой команды с учетом используемого вида адресации. Результаты, полученные с помощью отладчика, не являются объяснением, а только должны подтверждать ваши объяснения.

### **Ход работы.**

1. Получен 4 вариант с исходными данными для выполнения 2 лабораторной работы.

2. Протранслировала программу с созданием файла диагностических сообщений. Ниже представлены обнаруженные ошибки, которые были закомментированы.

```
Cross-reference INTEL.CRT:
lr2.asm(41): error A2052: Improper operand type
lr2.asm(48): warning A4031: Operand types must match
lr2.asm(52): warning A4031: Operand types must match
lr2.asm(53): error A2055: Illegal register value
lr2.asm(58): error A2046: Multiple base registers
lr2.asm(59): error A2047: Multiple index registers
lr2.asm(66): error A2006: Phase error between passes

49898 + 461459 Bytes symbol space free

2 Warning Errors
5 Severe Errors
```

lr2.asm(41): error A2052: Improper operand type

```
mov mem3,[bx]
```

Машинные команды не могут манипулировать одновременно двумя операндами, находящимися в оперативной памяти, то есть в команде только 1 операнд может указывать на ячейку памяти, другой операнд должен быть либо регистром, либо непосредственным значением.

lr2.asm(48): warning A4031: Operand types must match

```
mov cx,vec2[di]
```

Ошибка из-за разного типа операндов:

cx – это слово, а vec2[di] – это размерность 1 байт

lr2.asm(52): warning A4031: Operand types must match

```
mov cx,matr[bx][di]
```

Ошибка из-за разного типа операндов:

cx – это слово, а matr[bx][di] – это размерность 1 байт

lr2.asm(53): error A2055: Illegal register value

```
mov ax,matr[bx*4][di]
```

Регистр bx не может подвергаться масштабированию при индексной адресации.

lr2.asm(58): error A2046: Multiple base register

```
mov ax,matr[bp+bx]
```

Оба регистра в данном случае являются базовыми, а в косвенной адресации с индексированием адрес должен браться в виде суммы адресов, находящихся в базовом и индексном регистрах соответственно.

lr2.asm(59): error A2047: Multiple index register

```
mov ax,matr[bp+di+si]
```

В данной строке два индексных регистра и один базовый, а в непосредственной адресации с базированием и индексированием должна браться сумма базового индексного регистра, к которому добавляется смещение.

lr2.asm(66): error A2006: Phase error between passes

Main ENDP

Ошибка говорит о том, что в функции Main допущены ошибки.

3. Снова протранслирована программа и скомпонован загрузочный модуль.

Трансляция программы после исправления ошибок

```
D:\>masm lr2_cor.asm
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lr2_cor.OBJ]: lr2_cor
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

49958 + 461399 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

4. Программа выполнена в пошаговом режиме под управлением отладчика с фиксацией содержимого используемых регистров и ячеек памяти до и после выполнения команды.

lr2.asm

Адрес команды	Символический код команды	16-ричный код команды	Изменяемые данные	
			до	после
0000	PUSH DS	1E	STACK+0=0000 IP = 0000 SP=0018	STACK+0=19F5 IP = 0001 SP=0016
0001	SUB AX, AX	2BC0	AX=0000 IP = 0001	AX=0000 IP = 0003
0003	PUSH AX	50	STACK+0=19F5 STACK+2=0000 SP=0016 IP = 0003	STACK+0=0000 STACK+2=19F5 SP=0014 IP = 0004
0004	MOV AX,1A07	B8071A	AX = 0000 IP = 0004	AX =1A07 IP = 0007

0007	MOV DS,AX	8ED8	DS=19F5 IP = 0007	DS=1A07 IP = 0009
0009	MOV AX,01F4	B8F401	AX = 1A07 IP = 0009	AX = 01F4 IP = 000C
000C	MOV CX,AX	8BC8	CX=0096 IP=000C	CX=01F4 IP=000E
000E	MOV BL,24	B324	BX=0000 IP=000E	BX=0024 IP=0010
0010	MOV BH,CE	B7CE	BX=0024 IP=0010	BX=CE24 IP=0012
0012	MOV [0002],FFCE	C7060200CEFF	IP=0012	IP=0018
0018	MOV BX,0006	BB0600	BX=CE24 IP=0018	BX=0006 IP=001B
001B	MOV [0000],AX	A30000	IP=001B	IP=001E
001E	MOV AL,[BX]	8A07	AX=01F4 IP=001E	AX=010C IP=0020
0020	MOV AL,[BX+03]	8A4703	AX=010C IP=0020	AX=0109 IP=0023
0023	MOV CX, [BX+03]	8B4F03	CX= 01F4 IP = 0023	CX = 0509 IP= 0026
0026	MOV DI, 0002	BF0200	DI= 0000 IP= 0026	DI= 0002 IP= 0029
0029	MOV AL, [000E+DI]	8A850E00	AX= 0109 IP = 0029	AX= 012B IP= 002D

002D	MOV BX, 0003	BB0300	BX = 0006 IP = 002D	BX = 0003 IP = 0030
0030	MOV AL, [0016+BX+DI]	8A811600	AX = 012B IP = 0030	AX = 01F9 IP = 0034
0034	MOV AX, 1A07	B8071A	AX = 01F9 IP = 0034	AX = 1A07 IP = 0037
0037	MOV ES, AX	8EC0	ES = 19F5 IP = 0037	ES = 1A07 IP = 0039
0039	MOV AX, ES:[BX]	268B07	AX = 1A07 IP = 0039	AX = 00FF IP = 003C
003C	MOV AX, 0000	B80000	AX = 00FF IP = 003C	AX = 0000 IP = 003F
003F	MOV ES, AX	8EC0	ES = 1A07 IP = 003F	ES = 0000 IP = 0041
0041	PUSH DS	1E	IP = 0041 SP = 0014 STACK + 0 = 0000 STACK + 2 = 19F5 STACK + 4 = 0000	IP = 0042 SP = 0012 STACK + 0 = 1A07 STACK + 2 = 0000 STACK + 4 = 19F5
0042	POP ES	07	SP = 0012	SP = 0014

			ES=0000 IP= 0042 STACK +0 = 1A07 STACK +2 = 0000 STACK +4 =19F5	ES=1A07 IP= 0043 STACK +0 = 0000 STACK +2 = 19F5 STACK +4 =0000
0043	MOV CX, ES:[BX—01]	268B4FFF	CX = 0509 IP = 0043	CX= FFCE IP= 0047
0047	XCHG AX, CX	91	AX = 0000 CX = FFCE IP=0047	AX = FFCE CX = 0000 IP=0048
0048	MOV DI, 0002	BF0200	IP = 0048 DI=0002	IP = 004B DI=0002
004B	MOV ES:[BX+DI], AX	268901	IP = 004B	IP = 004E
004E	MOV BP, SP	8BEC	IP = 004E BP = 0000	IP = 0050 BP = 0014
0050	PUSH [0000]	FF360000	IP = 0050 SP=0014 STACK +0 = 0000 STACK +2 = 19F5 STACK +4	IP = 0054 SP=0012 STACK +0 = 01F4 STACK +2 = 0000 STACK +4

			=0000	=19F5
0054	PUSH [0002]	FF360200	IP = 0054 SP = 0012 STACK +0 = 01F4 STACK +2 = 0000 STACK +4 = 19F5 STACK +6 = 0000	IP = 0058 SP = 0010 STACK +0 = FFCE STACK +2 = 01F4 STACK +4 = 0000 STACK +6 = 19F5
0058	MOV BP, SP	8BEC	IP = 0058 BP = 0014	IP = 005A BP = 0010
005A	MOV DX, [BP+02]	8B5602	IP = 005A DX = 0000	IP = 005D DX = 01F4
005D	RET Far 0002	CA0200	IP = 005D SP = 0010 CS=1A0A STACK +0 = FFCE STACK +2 = 01F4 STACK +4 = 0000 STACK +6 = 19F5	IP = FFCE SP= 0016 CS=01F4 STACK +0 = 19F5 STACK +2 = 0000 STACK +4 = 0000 STACK +6 = 0000



**Выводы.**

В ходе выполнения лабораторной работы были изучены основные навыки работы с режимами адресации на языке программирования Ассемблер.

## Приложение А. Код программ.

Имя файла: lr2.asm

```
EOL EQU '$'
ind EQU 2
n1 EQU 500
n2 EQU -50
; Стек программы
AStack SEGMENT STACK
    DW 12 DUP(?)
AStack ENDS
; Данные программы
DATA SEGMENT
; Директивы описания данных
mem1 DW 0
mem2 DW 0
mem3 DW 0
vec1 DB 12,11,10,9,5,6,7,8
vec2 DB -40,-50,40,50,-20,-30,20,30
matr DB 5,6,7,8,-8,-7,-6,-5,1,2,3,4,-4,-3,-2,-1
DATA ENDS
; Код программы
CODE SEGMENT
    ASSUME CS:CODE, DS:DATA, SS:AStack
; Головная процедура
Main PROC FAR
    push DS
    sub AX,AX
    push AX
    mov AX,DATA
    mov DS,AX
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ НА УРОВНЕ СМЕЩЕНИЙ
; Регистровая адресация
    mov ax,n1
    mov cx,ax
    mov bl,EOL
```

```

    mov bh,n2
; Прямая адресация
    mov mem2,n2
    mov bx,OFFSET vec1
    mov mem1,ax
; Косвенная адресация
    mov al,[bx]
    mov mem3,[bx]
; Базированная адресация
7
    mov al,[bx]+3
    mov cx,3[bx]
; Индексная адресация
    mov di,ind
    mov al,vec2[di]
    mov cx,vec2[di]
; Адресация с базированием и индексированием
    mov bx,3
    mov al,matr[bx][di]
    mov cx,matr[bx][di]
    mov ax,matr[bx*4][di]
; ПРОВЕРКА РЕЖИМОВ АДРЕСАЦИИ С УЧЕТОМ СЕГМЕНТОВ
; Переопределение сегмента
; ----- вариант 4
    mov bp,sp
    mov ax,matr[bp+bx]
    mov ax,matr[bp+di+si]
; Использование сегмента стека
    push mem1
    push mem2
    mov bp,sp
    mov dx,[bp]+2
    ret 2
Main ENDP
CODE ENDS
    END Main

```

## Приложение Б. Листинг успешной трансляции программ.

Имя файла: lr2.lst

Microsoft (R) Macro Assembler Version 5.10  
10/8/22 15:26:16

Page

1-1

```
= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50
                ; PŸC,PµPe PiCṪPsPiCṪP°PjPjC<
0000          AStack SEGMENT STACK
0000 000C[      DW 12 DUP(?)
                ????
                ]

0018          AStack ENDS
                ; P°P°PSPSC<Pµ PiCṪPsPiCṪP°PjPjC<
0000          DATA SEGMENT
                ; P°PṪCṪPµPeC,PḔPIC< PSpṪPḔCṪP°PSPḔCṪ PrP°PSPSC
                <C...
0000 0000          mem1 DW 0
0002 0000          mem2 DW 0
0004 0000          mem3 DW 0
0006 0C 0B 0A 09 05 06 vec1 DB 12,11,10,9,5,6,7,8
                07 08
000E D8 CE 28 32 EC E2 vec2 DB -40,-50,40,50,-20,-30,20,30
                14 1E
0016 05 06 07 08 F8 F9 matr DB 5,6,7,8,-8,-7,-6,-5,1,2,3,4,-4,-
3,-2,-1
                FA FB 01 02 03 04
                FC FD FE FF

0026          DATA ENDS
                ; PḔPsPr PiCṪPsPiCṪP°PjPjC<
0000          CODE SEGMENT
                ASSUME CS:CODE, DS:DATA, SS:AStack
                ; P°PsP»PsPIPSP°CṪ PiCṪPsC†PµPrCṪCṪP°
0000          Main PROC FAR
0000 1E          push DS
0001 2B C0          sub AX,AX
0003 50          push AX
0004 B8 ---- R      mov AX,DATA
0007 8E D8          mov DS,AX
                ; PµP PhP'P•P PḔPh P P•P-PḔPḔPhP' PhP°P P•PŸPhP
                |PḔPḔ PḔPh PJP PhP'PḔP• PŸPḔP•PḔP•PḔPḔP™
                ; P PµPiPḔCṪC,CṪPsPIP°CṪ P°PrCṪPµCṪP°C†PḔCṪ
0009 B8 01F4          mov ax,n1
```

```

000C  8B C8          mov cx,ax
000E  B3 24          mov bl,EOL
0010  B7 CE          mov bh,n2
                ; PµCṪCṪPjP°CṪ P°PrCṪPµCṪP°C†PḔCṪ
0012  C7 06 0002 R FFCE  mov mem2,n2
0018  BB 0006 R      mov bx,OFFSET vec1
001B  A3 0000 R      mov mem1,ax
                ; PḄPsCṪPIPµPSPSP°CṪ P°PrCṪPµCṪP°C†PḔCṪ
001E  8A 07          mov al,[bx]
                mov mem3,[bx]
lr2.asm(41): error A2052: Improper operand type
                ; P`P°P·PḔCṪPsPIP°PSPSP°CṪ P°PrCṪPµCṪP°C†PḔCṪ
                7
lr2.asm(43): warning A4001: Extra characters on line
0020  8A 47 03          mov al,[bx]+3
0023  8B 4F 03          mov cx,3[bx]

```

Microsoft (R) Macro Assembler Version 5.10  
10/8/22 15:26:16

Page

1-2

```

                ; PṀPSPPrPµPeCṪPSP°CṪ P°PrCṪPµCṪP°C†PḔCṪ
0026  BF 0002          mov di,ind
0029  8A 85 000E R      mov al,vec2[di]
002D  8B 8D 000E R      mov cx,vec2[di]
lr2.asm(49): warning A4031: Operand types must match
                ; PḡPrCṪPµCṪP°C†PḔCṪ CṪ P±P°P·PḔCṪPsPIP°PSPḔPµP
                j PḔ PḔPSPPrPµPeCṪPḔCṪPsPIP°PSPḔPµPj
0031  BB 0003          mov bx,3
0034  8A 81 0016 R      mov al,matr[bx][di]
0038  8B 89 0016 R      mov cx,matr[bx][di]
lr2.asm(53): warning A4031: Operand types must match
003C  8B 85 0022 R      mov ax,matr[bx*4][di]
lr2.asm(54): error A2055: Illegal register value
                ; PµP PḡP'P·P PḄPḡ P P·P-PṀPḄPḡP' PḡP" P P·PḲPḡP
                |PṀPṀ PḲ PJPSP·PḲPḡPḄ PḲP·P"PḄP·PḲPḲPḡP'
                ; PµPµCṪPµPsPiCṪPµPrPµP»PµPSPḔPµ CṪPµPiPjPµPSC,
                P°
                ; ----- PIP°CṪPḔP°PSC, 4
0040  8B EC          mov bp,sp
0042  3E: 8B 86 0016 R      mov ax,matr[bp+bx]
lr2.asm(59): error A2046: Multiple base registers
0047  3E: 8B 83 0016 R      mov ax,matr[bp+di+si]
lr2.asm(60): error A2047: Multiple index registers
                ; PṀCṪPiPsP»CṪP·PsPIP°PSPḔPµ CṪPµPiPjPµPSC,P° C
                ṪC,PµPeP°
004C  FF 36 0000 R      push mem1
0050  FF 36 0002 R      push mem2
0054  8B EC          mov bp,sp

```

```

0056  8B 56 02                mov dx,[bp]+2
0059  CA 0002                ret 2
005C                                Main ENDP
lr2.asm(67): error A2006: Phase error between passes
005C                                CODE ENDS
                                END Main

```

Microsoft (R) Macro Assembler Version 5.10  
 10/8/22 15:26:16

Symbols-1

#### Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK	. . . . .	0018	PARA	STACK
CODE	. . . . .	005C	PARA	NONE
DATA	. . . . .	0026	PARA	NONE

#### Symbols:

	N a m e	Type	Value	Attr
EOL	. . . . .	NUMBER	0024	
IND	. . . . .	NUMBER	0002	
MAIN	. . . . .	F PROC	0000	CODE Length = 005C
MATR	. . . . .	L BYTE	0016	DATA
MEM1	. . . . .	L WORD	0000	DATA
MEM2	. . . . .	L WORD	0002	DATA
MEM3	. . . . .	L WORD	0004	DATA
N1	. . . . .	NUMBER	01F4	
N2	. . . . .	NUMBER	-0032	
VEC1	. . . . .	L BYTE	0006	DATA
VEC2	. . . . .	L BYTE	000E	DATA
@CPU	. . . . .	TEXT	0101h	
@FILENAME	. . . . .	TEXT	lr2	
@VERSION	. . . . .	TEXT	510	

69 Source Lines  
 69 Total Lines  
 19 Symbols

47842 + 461465 Bytes symbol space free

### 3 Warning Errors

## 5 Severe Errors

Имя файла: lr2.lst (исправленный код)

Microsoft (R) Macro Assembler Version 5.10

10/8/22 15:19:30

Page

1-1

```

= 0024          EOL EQU '$'
= 0002          ind EQU 2
= 01F4          n1 EQU 500
=-0032          n2 EQU -50
                ; Ĝ;Ñ,ĜµĜ° Ĝ;Ñ€Ĝ¾Ĝ³Ñ€Ĝ°Ĝ¼Ĝ¼Ñ<

0000            Astack SEGMENT STACK
0000    000C[           DW 12 DUP(?)
        ????

]

0018            Astack ENDS
                ; Ĝ"Ĝ°Ĝ½Ĝ½Ñ< Ĝµ Ĝ;Ñ€Ĝ¾Ĝ³Ñ€Ĝ°Ĝ¼Ĝ¼Ñ<

0000            DATA SEGMENT
                ; Ĝ"Ĝ,Ñ€ĜµĜ°Ñ,Ĝ,Ĝ²Ñ< Ĝ¾Ĝ;Ĝ,Ñ□Ĝ°Ĝ½Ĝ,Ñ□ Ĝ´Ĝ°Ĝ½Ĝ½Ñ
<Ñ...

0000    0000            mem1 DW 0
0002    0000            mem2 DW 0
0004    0000            mem3 DW 0
0006    0C 0B 0A 09 05 06 vec1 DB 12,11,10,9,5,6,7,8
        07 08
000E    D8 CE 28 32 EC E2 vec2 DB -40,-50,40,50,-20,-30,20,30
        14 1E
0016    05 06 07 08 F8 F9 matr DB 5,6,7,8,-8,-7,-6,-5,1,2,3,4,-4,-
3,-2,-1
        FA FB 01 02 03 04
        FC FD FE FF

0026            DATA ENDS
                ; ĜšĜ¾Ĝ´ Ĝ;Ñ€Ĝ¾Ĝ³Ñ€Ĝ°Ĝ¼Ĝ¼Ñ<

0000            CODE SEGMENT
                ASSUME CS:CODE, DS:DATA, SS:Astack
                ; Ĝ"Ĝ¾Ĝ»Ĝ¾Ĝ²Ĝ½Ĝ°Ñ□ Ĝ;Ñ€Ĝ¾Ñ†ĜµĜ´ÑfÑ€Ĝ°

0000            Main PROC FAR
0000    1E             push DS
0001    2B C0          sub AX,AX
0003    50             push AX

```

```

0004 B8 ---- R      mov AX,DATA
0007 8E D8          mov DS,AX
; ĞŸĜ Ğ□Ĝ'Ĝ•Ĝ ĞšĜ□ Ğ Ğ•Ĝ-Ĝ~ĜœĜ□Ĝ' Ğ□Ĝ"Ĝ Ğ•Ĝ;Ĝ□Ĝ
|Ĝ~Ĝ~ Ğ□Ĝ□ ĞŁĜ Ğ□Ĝ'Ĝ□Ĝ• Ğ;ĜœĜ•Ĝ©Ĝ•Ĝ□Ĝ~Ĝ™
; Ğ ĞµĜ³Ĝ,Ń□Ń,ŃĖĜ¼Ĝ²Ĝ°Ń□ Ğ°Ĝ'ŃĖĜµŃ□Ĝ°Ń+Ĝ,Ń□
0009 B8 01F4        mov ax,n1
000C 8B C8          mov cx,ax
000E B3 24          mov bl,EOL
0010 B7 CE          mov bh,n2
; ĞŸŃĖŃ□Ĝ¼Ĝ°Ń□ Ğ°Ĝ'ŃĖĜµŃ□Ĝ°Ń+Ĝ,Ń□
0012 C7 06 0002 R FFCE mov mem2,n2
0018 BB 0006 R      mov bx,OFFSET vec1
001B A3 0000 R      mov mem1,ax
; ĞšĜ¼Ń□Ĝ²ĜµĜ¼Ĝ¼Ĝ°Ń□ Ğ°Ĝ'ŃĖĜµŃ□Ĝ°Ń+Ĝ,Ń□
001E 8A 07          mov al,[bx]
;mov mem3,[bx]
; Ğ'Ĝ°Ĝ•Ĝ,ŃĖĜ¼Ĝ²Ĝ°Ĝ¼Ĝ¼Ĝ°Ń□ Ğ°Ĝ'ŃĖĜµŃ□Ĝ°Ń+Ĝ,Ń□
;7
0020 8A 47 03      mov al,[bx]+3
0023 8B 4F 03      mov cx,3[bx]

```

Microsoft (R) Macro Assembler Version 5.10

10/8/22 15:19:30

Page

1-2

```

; Ğ~Ĝ¼Ĝ'ĜµĜ°Ń□Ĝ¼Ĝ°Ń□ Ğ°Ĝ'ŃĖĜµŃ□Ĝ°Ń+Ĝ,Ń□
0026 BF 0002        mov di,ind
0029 8A 85 000E R   mov al,vec2[di]
;mov cx,vec2[di]
; Ğ□Ĝ'ŃĖĜµŃ□Ĝ°Ń+Ĝ,Ń□ Ń□ Ğ±Ĝ°Ĝ•Ĝ,ŃĖĜ¼Ĝ²Ĝ°Ĝ¼Ĝ,ĜµĜ
¼ Ğ, Ğ,Ĝ¼Ĝ'ĜµĜ°Ń□Ĝ,ŃĖĜ¼Ĝ²Ĝ°Ĝ¼Ĝ,ĜµĜ¼
002D BB 0003        mov bx,3
0030 8A 81 0016 R   mov al,matr[bx][di]
;mov cx,matr[bx][di]
;mov ax,matr[bx*4][di]
; ĞŸĜ Ğ□Ĝ'Ĝ•Ĝ ĞšĜ□ Ğ Ğ•Ĝ-Ĝ~ĜœĜ□Ĝ' Ğ□Ĝ"Ĝ Ğ•Ĝ;Ĝ□Ĝ
|Ĝ~Ĝ~ Ğ; ĞŁĜšĜ•ĜĉĜ□Ĝœ Ğ;Ğ•Ĝ"ĜœĜ•Ĝ□ĜĉĜ□Ĝ'
; ĞŸĜµŃĖĜµĜ¼Ĝ;ŃĖĜµĜ'ĜµĜ»ĜµĜ¼Ĝ,Ĝµ Ń□ĜµĜ³Ĝ¼ĜµĜ¼Ń,
Ĝ°
; ----- Ğ²Ĝ°ŃĖĜ,Ĝ°Ĝ¼Ń, 4
0034 8B EC          mov bp,sp
;mov ax,matr[bp+bx]
;mov ax,matr[bp+di+si]
; Ğ~Ń□Ĝ;Ĝ¼Ĝ»ŃĖĜ•Ĝ¼Ĝ²Ĝ°Ĝ¼Ĝ,Ĝµ Ń□ĜµĜ³Ĝ¼ĜµĜ¼Ń,Ĝ° Ń
Ń,ĜµĜ°Ĝ°
0036 FF 36 0000 R   push mem1
003A FF 36 0002 R   push mem2
003E 8B EC          mov bp,sp
0040 8B 56 02        mov dx,[bp]+2

```



```

0043  CA 0002          ret 2
0046                      Main ENDP
0046                      CODE ENDS
                        END Main

```

```

Microsoft (R) Macro Assembler Version 5.10
10/8/22 15:19:30

```

Symbols-1

Segments and Groups:

Class	N a m e	Length	Align	Combine
ASTACK	. . . . .	0018	PARA	STACK
CODE	. . . . .	0046	PARA	NONE
DATA	. . . . .	0026	PARA	NONE

Symbols:

	N a m e	Type	Value	Attr
EOL	. . . . .	NUMBER	0024	
IND	. . . . .	NUMBER	0002	
MAIN	. . . . .	F PROC	0000	CODE Length = 0046
MATR	. . . . .	L BYTE	0016	DATA
MEM1	. . . . .	L WORD	0000	DATA
MEM2	. . . . .	L WORD	0002	DATA
MEM3	. . . . .	L WORD	0004	DATA
N1	. . . . .	NUMBER	01F4	
N2	. . . . .	NUMBER	-0032	
VEC1	. . . . .	L BYTE	0006	DATA
VEC2	. . . . .	L BYTE	000E	DATA
@CPU	. . . . .	TEXT	0101h	
@FILENAME	. . . . .	TEXT	lr2_cor	
@VERSION	. . . . .	TEXT	510	

```

69 Source  Lines
69 Total   Lines
19 Symbols

```

47898 + 461409 Bytes symbol space free

0 Warning Errors  
0 Severe Errors