



# **Bölüm 7: Akış Kontrol**

## **Mikroişlemciler**



# Program Akış Kontrolü

- Koşulsuz atlamalar (*jumps*)
- Koşullu atlamalar (*jumps*)
  - Tek bir bayrağı test eden atlama komutları
  - İşaretli sayılar için atlama komutları
  - İşaretsiz sayılar için atlama komutları
- Döngüler (*loops*)



# Koşulsuz Atlamalar

- JMP (Jump): program içinde kontrolü başka bir noktaya aktarır.
  - JMP etiket
- Etiket tanımlamak için adı yazılır ve sonuna ":" eklenir.
- Etiket herhangi bir karakter kombinasyonu olabilir,
  - ancak bir sayı ile başlayamaz.
- Etiket, ayrı bir satırda veya başka bir komutun önünde tanımlanabilir.
- JMP, kontrolü hem ileri hem de geri yönlendirebilir.
- Mevcut kod segmenti içinde (65,535 bayt) herhangi bir yere atlayabilir.



# Koşulsuz Atlamalar

```
org    100h
mov    ax, 5           ; set ax to 5.
mov    bx, 2           ; set bx to 2.
jmp    calc            ; go to 'calc'.
back:  jmp stop        ; go to 'stop'.
calc:
add    ax, bx          ; add bx to ax.
jmp    back            ; go 'back'.
stop:
ret                                ; return to operating system.
```



# Tek Bir Bayrağı Test Eden Atlama Komutları

Instruction	Description	Condition
JZ , JE	Jump if Zero (Equal).	ZF = 1
JC , JB, JNAE	Jump if Carry (Below, Not Above Equal).	CF = 1
JS	Jump if Sign.	SF = 1
JO	Jump if Overflow.	OF = 1
JPE, JP	Jump if Parity Even.	PF = 1
JNZ , JNE	Jump if Not Zero (Not Equal).	ZF = 0
JNC , JNB, JAE	Jump if Not Carry (Not Below, Above Equal).	CF = 0
JNS	Jump if Not Sign.	SF = 0
JNO	Jump if Not Overflow.	OF = 0
JPO, JNP	Jump if Parity Odd (No Parity).	PF = 0



# Tek Bir Bayrağı Test Eden Atlama Komutları

- JZ, JE: Sıfıra eşitse atlama yap.
  - Koşul: ZF = 1 Zıt Komut: JNZ, JNE
- JC, JB, JNAE: Taşma durumunda atlama yap.
  - Koşul: CF = 1 Zıt Komut: JNC, JNB, JAE
- JS: Negatifse atlama yap.
  - Koşul: SF = 1 Zıt Komut: JNS
- JO: Taşma durumunda atlama yap.
  - Koşul: OF = 1 Zıt Komut: JNO
- JPE, JP: Çiftlik durumunda atlama yap.
  - Koşul: PF = 1 Zıt Komut: JPO



# Tek Bir Bayrağı Test Eden Atlama Komutları

- Atlama komutları sabit uzunluktadır (iki bayt).
- Bağlı konum (*Offset*) 1 baytta saklanır.
  - -128 bayt geriye veya 127 bayt ileriye atlama yapabilir.
- Değer her zaman işaretli bir sayıdır.
- JE, JZ; JNE, JNZ ile aynı makine koduna derlenir.
- JC, JB, JNAE; JNC, JNB, JAE ile aynı makine koduna derlenir.



# Tek Bir Bayrağı Test Eden Atlama Komutları

```
jnc a
```

```
jnb a
```

```
jae a
```

```
mov ax, 4
```

```
a: mov ax, 5
```

```
ret
```





# İşaretili Sayılar İçin Atlama Komutları

Instruction	Description	Condition
JE , JZ	Jump if Equal (=). Jump if Zero.	ZF = 1
JNE , JNZ	Jump if Not Equal (<>). Jump if Not Zero.	ZF = 0
JG , JNLE	Jump if Greater (>). Jump if Not Less or Equal (not <=).	ZF = 0 and SF = OF
JL , JNGE	Jump if Less (<). Jump if Not Greater or Equal (not >=).	SF <> OF
JGE , JNL	Jump if Greater or Equal (>=). Jump if Not Less (not <).	SF = OF
JLE , JNG	Jump if Less or Equal (<=). Jump if Not Greater (not >).	ZF = 1 or SF <> OF



# İşaretili Sayılar İçin Atlama Komutları

- JE, JZ: Eşitse atlama yap.
  - Koşul:  $ZF = 1$  Zıt Komut: JNE, JNZ
- JNE, JNZ: Eşit değilse atlama yap.
  - Koşul:  $ZF = 0$  Zıt Komut: JE, JZ
- JG, JNLE: Büyükse atlama yap.
  - Koşul:  $ZF = 0$  ve  $SF = OF$  Zıt Komut: JNG, JLE
- JL, JNGE: Küçükse atlama yap.
  - Koşul:  $SF \neq OF$  Zıt Komut: JNL, JGE
- JGE, JNL: Büyük veya eşitse atlama yap.
  - Koşul:  $SF = OF$  Zıt Komut: JNGE, JL
- JLE, JNG: Küçük veya eşitse atlama yap.
  - Koşul:  $ZF = 1$  veya  $SF \neq OF$  Zıt Komut: JNLE, JG



# İşaretili Sayılar İçin Atlama Komutları

- <> işareti eşit değil anlamına gelir.

```
mov ax, 5
```

```
mov bx, 5
```

```
cmp ax, bx
```

```
je equal_message
```

```
jmp not_equal_message
```

```
equal_message:           ; Eşitse yapılacak işlemler
```

```
jmp end_program
```

```
not_equal_message:      ; Eşit değilse yapılacak işlemler
```

```
end_program:
```



# İşaretsiz Sayılar İçin Atlama Komutları

Instruction	Description	Condition
JE , JZ	Jump if Equal (=). Jump if Zero.	ZF = 1
JNE , JNZ	Jump if Not Equal (<>). Jump if Not Zero.	ZF = 0
JA , JNBE	Jump if Above (>). Jump if Not Below or Equal (not <=).	CF = 0 and ZF = 0
JB , JNAE, JC	Jump if Below (<). Jump if Not Above or Equal (not >=). Jump if Carry.	CF = 1
JAE , JNB, JNC	Jump if Above or Equal (>=). Jump if Not Below (not <). Jump if Not Carry.	CF = 0
JBE , JNA	Jump if Below or Equal (<=). Jump if Not Above (not >).	CF = 1 or ZF = 1



# İşaretsiz Sayılar İçin Atlama Komutları

- JE, JZ: Eşitse atlama yap.
  - Koşul:  $ZF = 1$  Zıt Komut: JNE, JNZ
- JNE, JNZ: Eşit değilse atlama yap.
  - Koşul:  $ZF = 0$  Zıt Komut: JE, JZ
- JA, JNBE: Büyükse atlama yap.
  - Koşul:  $CF = 0$  ve  $ZF = 0$  Zıt Komut: JNA, JBE
- JB, JNAE, JC: Küçükse atlama yap.
  - Koşul:  $CF = 1$  Zıt Komut: JNB, JAE, JNC
- JAE, JNB, JNC: Büyük veya eşitse atlama yap.
  - Koşul:  $CF = 0$  Zıt Komut: JNAE, JB
- JBE, JNA: Küçük veya eşitse atlama yap.
  - Koşul:  $CF = 1$  veya  $ZF = 1$  Zıt Komut: JNBE, JA



# İşaretsiz Sayılar İçin Atlama Komutları

```
mov ax, 5
mov bx, 7
cmp ax, bx
ja  jump_above
jmp not_jump_above
jump_above:      ; ax büyükse yapılacak işlemler
jmp end_program
not_jump_above: ; ax küçükse veya eşitse yapılacak işlemler
end_program:
```



# CMP ve Atlama Komutları

- Sayısal değerleri karşılaştırmak için CMP (compare) komutu kullanılır.
- CMP komutu, SUB (çıkarma) komutunu gerçekleştirir.
- Örnek 1: 5 ve 2'yi karşılaştır,
  - $5 - 2 = 3$
  - Sonuç sıfır değil (Zero Bayrağına 0 atanır).
- Örnek 2: 7 ve 7'yi karşılaştır,
  - $7 - 7 = 0$
  - Sonuç sıfır! (Zero Bayrağına 1 atanır, JZ veya JE atlama yapar).



# CMP ve Atlama Komutları

```
include "emu8086.inc"
org    100h
mov    al, 25    ; set al to 25.
mov    bl, 10    ; set bl to 10.
cmp    al, bl    ; compare al - bl.
je     equal     ; jump if al = bl (zf = 1).
putc   'n'       ; if it gets here, then al <> bl,
jmp    stop      ; so print 'n', and jump to stop.
equal:          ; if gets here,
putc   'y'       ; then al = bl, so print 'y'.
stop:
ret      ; gets here no matter what.
```





# Döngüler (Loops)

Instruction	Operation And Jump Condition
LOOP	decrease cx, jump to label if cx not zero.
LOOPE	decrease cx, jump to label if cx not zero and equal (zf = 1).
LOOPNE	decrease cx, jump to label if cx not zero and not equal (zf = 0).
LOOPNZ	decrease cx, jump to label if cx not zero and zf = 0.
LOOPZ	decrease cx, jump to label if cx not zero and zf = 1.
JCXZ	jump to label if cx is zero.



# Döngüler (Loops)

- Döngüler, bir koşula bağlı olarak bir kod bloğunun tekrarlanmasını sağlar.
- LOOP: CX sıfır olmadığı sürece belirtilen etikete atlama yapar.
- LOOPE, LOOPZ:
  - CX sıfır olmadığı ve ZF = 1 olduğu sürece etikete atlama yapar.
- LOOPNE, LOOPNZ :
  - CX sıfır olmadığı ve ZF = 0 olduğu sürece etikete atlama yapar.
- JCXZ: CX sıfır olduğunda belirtilen etikete atlama yapar.



# Döngüler (Loops)

```
include "emu8086.inc"
org 100h
mov cx, 5 ; CX döngü tekrar sayısı 5 ata.
dongu:
    ; Döngü İçeriği
    loop dongu ; CX sıfır değilse dongu etiketine atla
jmp dur ; Döngü bittiğinde dur etiketine atla
dur:
ret
```



# Döngüler (Loops)

```
1  org 100h
2  mov bx, 0 ;Toplam adım.
3  mov cx, 5
4  k1:
5  add bx, 1
6  mov al, '1'
7  mov ah, 0eh
8  int 10h
9  push cx
10 mov cx, 5
```

```
11 k2:
12 add bx, 1
13 mov al, '2'
14 mov ah, 0eh
15 int 10h
16 loop k2 ; İç döngü.
17 pop cx
18 loop k1 ; Dış döngü.
19 ret
```



# Dizi Elemanları Toplamı

```
org 100h
mov cx, 5 ; eleman sayısı
mov al, 0 ; toplam al yazmacında tutulacak
mov bx, 0 ; bx indis olarak kullanılacak
next: add al, vector[bx] ; elemanları topla
inc bx ; sonraki eleman
loop next ; cx=0 olana kadar dön
mov m, al ; sonucu m değişkenine atar
ret
vector db 5, 4, 5, 2, 1
m db 0
```



# Dizi Elemanları Toplamı

emulator: calc-sum.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	19
DX	00	00
CS	07 00	
IP	01 00	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0100

```
07100: B9 185 |
07101: 05 005 *
07102: 00 000 NULL
07103: B0 176 //
07104: 00 000 NULL
07105: BB 187 |
07106: 00 000 NULL
07107: 00 000 NULL
07108: 02 002 @
07109: 87 135 ¢
0710A: 13 019 !!
0710B: 01 001 @
0710C: 43 067 C
0710D: E2 226 Γ
0710E: F9 249 ·
0710F: A2 162 ó
07110: 18 024 †
07111: 01 001 @
07112: C3 195 †
07113: 05 005 *
07114: 04 004 ♦
07115: 05 005 *
```

0700:0100

```
MOV CX, 00005h
MOV AL, 00h
MOV BX, 00000h
ADD AL, [BX] + 00113h
INC BX
LOOP 0108h
MOV [00118h], AL
RET
ADD AX, 00504h
ADD AL, [BX + DI]
ADD [BX + SI] + 09090h, 1
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Dizi Elemanları Toplamı

emulator: calc-sum.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	05
DX	00	00
CS	07 00	
IP	01 08	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 08

```
07100: B9 185 |  
07101: 05 005 *  
07102: 00 000 NULL  
07103: B0 176 |  
07104: 00 000 NULL  
07105: BB 187 |  
07106: 00 000 NULL  
07107: 00 000 NULL  
07108: 02 002 @  
07109: 87 135 ç  
0710A: 13 019 ::  
0710B: 01 001 @  
0710C: 43 067 C  
0710D: E2 226 Γ  
0710E: F9 249 ·  
0710F: A2 162 ó  
07110: 18 024 †  
07111: 01 001 @  
07112: C3 195 †  
07113: 05 005 *  
07114: 04 004 ♦  
07115: 05 005 *
```

07 00: 01 08

```
MOV CX, 00005h  
MOV AL, 00h  
MOV BX, 00000h  
ADD AL, [BX] + 00113h  
INC BX  
LOOP 0108h  
MOV [00118h], AL  
RET  
ADD AX, 00504h  
ADD AL, [BX + DI]  
ADD [BX + SI] + 09090h, 1  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...
```

screen source reset aux vars debug stack flags



# Dizi Elemanları Toplamı

emulator: calc-sum.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	05
BX	00	01
CX	00	05
DX	00	00
CS	0700	
IP	010D	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:010D

```
07100: B9 185 |  
07101: 05 005 *  
07102: 00 000 NULL  
07103: B0 176 |  
07104: 00 000 NULL  
07105: BB 187 |  
07106: 00 000 NULL  
07107: 00 000 NULL  
07108: 02 002 @  
07109: 87 135 ¢  
0710A: 13 019 !!  
0710B: 01 001 @  
0710C: 43 067 C  
0710D: E2 226 |  
0710E: F9 249 -  
0710F: A2 162 6  
07110: 18 024 †  
07111: 01 001 @  
07112: C3 195 †  
07113: 05 005 *  
07114: 04 004 ♦  
07115: 05 005 *
```

0700:010D

```
MOV CX, 00005h  
MOV AL, 00h  
MOV BX, 00000h  
ADD AL, [BX] + 00113h  
INC BX  
LOOP 0108h  
MOV [00118h], AL  
RET  
ADD AX, 00504h  
ADD AL, [BX + DI]  
ADD [BX + SI] + 09090h, 1  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...
```

screen source reset aux vars debug stack flags





# Dizi Elemanları Toplamı

emulator: calc-sum.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	09
BX	00	01
CX	00	04
DX	00	00
CS	07 00	
IP	01 0C	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0C

```
07100: B9 185 |  
07101: 05 005 *  
07102: 00 000 NULL  
07103: B0 176 |  
07104: 00 000 NULL  
07105: BB 187 |  
07106: 00 000 NULL  
07107: 00 000 NULL  
07108: 02 002 @  
07109: 87 135 ¢  
0710A: 13 019 !!  
0710B: 01 001 @  
0710C: 43 067 C  
0710D: E2 226 |  
0710E: F9 249 ·  
0710F: A2 162 ó  
07110: 18 024 †  
07111: 01 001 @  
07112: C3 195 †  
07113: 05 005 *  
07114: 04 004 ♦  
07115: 05 005 *
```

07 00: 01 0C

```
MOV CX, 00005h  
MOV AL, 00h  
MOV BX, 00000h  
ADD AL, [BX] + 00113h  
INC BX  
LOOP 0108h  
MOV [00118h], AL  
RET  
ADD AX, 00504h  
ADD AL, [BX + DI]  
ADD [BX + SI] + 09090h, 1  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...
```

screen source reset aux vars debug stack flags



# Dizi Elemanları Toplamı

emulator: calc-sum.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	0E
BX	00	02
CX	00	03
DX	00	00
CS	07 00	
IP	01 0C	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0C

```
07100: B9 185 |  
07101: 05 005 *  
07102: 00 000 NULL  
07103: B0 176 |  
07104: 00 000 NULL  
07105: BB 187 |  
07106: 00 000 NULL  
07107: 00 000 NULL  
07108: 02 002 @  
07109: 87 135 ¢  
0710A: 13 019 !!  
0710B: 01 001 @  
0710C: 43 067 C  
0710D: E2 226 |  
0710E: F9 249 ·  
0710F: A2 162 ó  
07110: 18 024 †  
07111: 01 001 @  
07112: C3 195 †  
07113: 05 005 *  
07114: 04 004 ♦  
07115: 05 005 *
```

07 00: 01 0C

```
MOV CX, 00005h  
MOV AL, 00h  
MOV BX, 00000h  
ADD AL, [BX] + 00113h  
INC BX  
LOOP 0108h  
MOV [00118h], AL  
RET  
ADD AX, 00504h  
ADD AL, [BX + DI]  
ADD [BX + SI] + 09090h, 1  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...
```

screen source reset aux vars debug stack flags



# Dizi Elemanları Toplamı

emulator: calc-sum.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	10
BX	00	03
CX	00	02
DX	00	00
CS	07 00	
IP	01 0C	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0C

```
07100: B9 185 |  
07101: 05 005 *  
07102: 00 000 NULL  
07103: B0 176 |  
07104: 00 000 NULL  
07105: BB 187 |  
07106: 00 000 NULL  
07107: 00 000 NULL  
07108: 02 002 @  
07109: 87 135 ¢  
0710A: 13 019 !!  
0710B: 01 001 @  
0710C: 43 067 C  
0710D: E2 226 |  
0710E: F9 249 ·  
0710F: A2 162 ó  
07110: 18 024 †  
07111: 01 001 @  
07112: C3 195 †  
07113: 05 005 *  
07114: 04 004 ♦  
07115: 05 005 *
```

07 00: 01 0C

```
MOV CX, 00005h  
MOV AL, 00h  
MOV BX, 00000h  
ADD AL, [BX] + 00113h  
INC BX  
LOOP 0108h  
MOV [00118h], AL  
RET  
ADD AX, 00504h  
ADD AL, [BX + DI]  
ADD [BX + SI] + 09090h, 1  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...
```

screen source reset aux vars debug stack flags



# Dizi Elemanları Toplamı

emulator: calc-sum.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	11
BX	00	04
CX	00	01
DX	00	00
CS	0700	
IP	010C	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:010C

```
07100: B9 185 |
07101: 05 005 *
07102: 00 000 NULL
07103: B0 176 //
07104: 00 000 NULL
07105: BB 187 7
07106: 00 000 NULL
07107: 00 000 NULL
07108: 02 002 @
07109: 87 135 ¢
0710A: 13 019 !!
0710B: 01 001 @
0710C: 43 067 C
0710D: E2 226 7
0710E: F9 249 .
0710F: A2 162 6
07110: 18 024 †
07111: 01 001 @
07112: C3 195 †
07113: 05 005 *
07114: 04 004 ♦
07115: 05 005 *
```

0700:010C

```
MOV CX, 00005h
MOV AL, 00h
MOV BX, 00000h
ADD AL, [BX] + 00113h
INC BX
LOOP 0108h
MOV [00118h], AL
RET
ADD AX, 00504h
ADD AL, [BX + DI]
ADD [BX + SI] + 09090h, 1
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Dizi Elemanları Toplamı

emulator: calc-sum.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	11
BX	00	05
CX	00	00
DX	00	00
CS	07 00	
IP	0112	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	07 00	
ES	07 00	

0700:0112

```
07100: B9 185 |
07101: 05 005 *
07102: 00 000 NULL
07103: B0 176 //
07104: 00 000 NULL
07105: BB 187 7
07106: 00 000 NULL
07107: 00 000 NULL
07108: 02 002 @
07109: 87 135 ¸
0710A: 13 019 !!
0710B: 01 001 @
0710C: 43 067 C
0710D: E2 226 7
0710E: F9 249 .
0710F: A2 162 6
07110: 18 024 †
07111: 01 001 @
07112: C3 195 †
07113: 05 005 *
07114: 04 004 ♦
07115: 05 005 *
```

0700:0112

```
MOV CX, 00005h
MOV AL, 00h
MOV BX, 00000h
ADD AL, [BX] + 00113h
INC BX
LOOP 0108h
MOV [00118h], AL
RET
ADD AX, 00504h
ADD AL, [BX + DI]
ADC [BX + SI] + 09090h, 1
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Ekranaya Hello Yazdırma

```
jmp    start
msg db 'Hello', 0
start:
    mov    si, 0            ; source index'i ayarla
next_char:
    mov    al, msg[si]     ; yazdırılacak karakteri al
    cmp    al, 0           ; 0 ile karşılaştır
    je     stop           ; 0 ise yazmayı durdur
    mov    ah, 0eh        ; karakteri yazdır
    int    10h
    inc    si             ; source index'i güncelle
    jmp    next_char      ; diğer karakteri yazdırmaya geç
stop:  mov    ah, 0        ; klavye tuşuna basılmasını bekle
    int    16h
```



# Ekranaya Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0000

01000:	EB	235	δ
01001:	06	006	↑
01002:	48	072	H
01003:	65	101	e
01004:	6C	108	l
01005:	6C	108	l
01006:	6F	111	o
01007:	00	000	NULL
01008:	BE	190	↓
01009:	00	000	NULL
0100A:	00	000	NULL
0100B:	8A	138	è
0100C:	44	068	D
0100D:	02	002	2
0100E:	3C	060	<
0100F:	00	000	NULL
01010:	74	116	t
01011:	07	007	BEEP
01012:	B4	180	↓
01013:	0E	014	µ
01014:	CD	205	=
01015:	10	016	▶

0100:0000

```
JMP 08h
DEC AX
DB 65h
INSB
INSB
OUTSW
ADD [BP] + 00000h, BH
MOV AL, [SI] + 02h
CMP AL, 00h
JZ 019h
MOV AH, 0Eh
INT 010h
INC SI
JMP 0Bh
MOV AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Ekranaya Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0008	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0008

01000:	EB	235	δ
01001:	06	006	↑
01002:	48	072	H
01003:	65	101	e
01004:	6C	108	l
01005:	6C	108	l
01006:	6F	111	o
01007:	00	000	NULL
01008:	BE	190	↓
01009:	00	000	NULL
0100A:	00	000	NULL
0100B:	8A	138	è
0100C:	44	068	D
0100D:	02	002	ø
0100E:	3C	060	<
0100F:	00	000	NULL
01010:	74	116	t
01011:	07	007	BEEP
01012:	B4	180	↓
01013:	0E	014	ß
01014:	CD	205	=
01015:	10	016	▶

0100:0008

```
MOU SI, 00000h
MOU AL, [SI] + 02h
CMP AL, 00h
JZ 019h
MOU AH, 0Eh
INT 010h
INC SI
JMP 0Bh
MOU AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags





# Ekranaya Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	000B	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:000B

01000:	EB	235	δ
01001:	06	006	↑
01002:	48	072	H
01003:	65	101	e
01004:	6C	108	l
01005:	6C	108	l
01006:	6F	111	o
01007:	00	000	NULL
01008:	BE	190	↓
01009:	00	000	NULL
0100A:	00	000	NULL
0100B:	8A	138	è
0100C:	44	068	D
0100D:	02	002	è
0100E:	3C	060	<
0100F:	00	000	NULL
01010:	74	116	t
01011:	07	007	BEEP
01012:	B4	180	↓
01013:	0E	014	µ
01014:	CD	205	=
01015:	10	016	▶

0100:000B

```
MOU SI, 00000h
MOU AL, [SI] + 02h
CMP AL, 00h
JZ 019h
MOU AH, 0Eh
INT 010h
INC SI
JMP 0Bh
MOU AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Ekranaya Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	48
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	000E	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:000E

01000:	EB	235	δ
01001:	06	006	↑
01002:	48	072	H
01003:	65	101	e
01004:	6C	108	l
01005:	6C	108	l
01006:	6F	111	o
01007:	00	000	NULL
01008:	BE	190	↓
01009:	00	000	NULL
0100A:	00	000	NULL
0100B:	8A	138	è
0100C:	44	068	D
0100D:	02	002	ø
0100E:	3C	060	<
0100F:	00	000	NULL
01010:	74	116	t
01011:	07	007	BEEP
01012:	B4	180	↓
01013:	0E	014	¶
01014:	CD	205	=
01015:	10	016	▶

0100:000E

```
MOV SI, 00000h
MOV AL, [SI] + 02h
CMP AL, 00h
JZ 019h
MOV AH, 0Eh
INT 010h
INC SI
JMP 0Bh
MOV AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Ekran Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	48
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0010	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0010

01000:	EB	235	δ
01001:	06	006	↑
01002:	48	072	H
01003:	65	101	e
01004:	6C	108	l
01005:	6C	108	l
01006:	6F	111	o
01007:	00	000	NULL
01008:	BE	190	↓
01009:	00	000	NULL
0100A:	00	000	NULL
0100B:	8A	138	è
0100C:	44	068	D
0100D:	02	002	0
0100E:	3C	060	<
0100F:	00	000	NULL
01010:	74	116	t
01011:	07	007	BEEP
01012:	B4	180	↓
01013:	0E	014	¶
01014:	CD	205	=
01015:	10	016	▶

0100:0010

```
MOV SI, 00000h
MOV AL, [SI] + 02h
CMP AL, 00h
JZ 019h
MOV AH, 0Eh
INT 010h
INC SI
JMP 0Bh
MOV AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Ekranaya Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	48
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0012	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0012

```
01000: EB 235 6
01001: 06 006
01002: 48 072 H
01003: 65 101 e
01004: 6C 108 l
01005: 6C 108 l
01006: 6F 111 o
01007: 00 000 NULL
01008: BE 190 j
01009: 00 000 NULL
0100A: 00 000 NULL
0100B: 8A 138 è
0100C: 44 068 D
0100D: 02 002
0100E: 3C 060 <
0100F: 00 000 NULL
01010: 74 116 t
01011: 07 007 BEEP
01012: B4 180 |
01013: 0E 014 j
01014: CD 205 =
01015: 10 016 ▶
```

0100:0012

```
MOV SI, 00000h
MOV AL, [SI] + 02h
CMP AL, 00h
JZ 019h
MOV AH, 0Eh
INT 010h
INC SI
JMP 0Bh
MOV AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Ekranaya Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	0E	48
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0014	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0014

01000:	EB	235	δ
01001:	06	006	↑
01002:	48	072	H
01003:	65	101	e
01004:	6C	108	l
01005:	6C	108	l
01006:	6F	111	o
01007:	00	000	NULL
01008:	BE	190	↓
01009:	00	000	NULL
0100A:	00	000	NULL
0100B:	8A	138	è
0100C:	44	068	D
0100D:	02	002	2
0100E:	3C	060	<
0100F:	00	000	NULL
01010:	74	116	t
01011:	07	007	BEEP
01012:	B4	180	↓
01013:	0E	014	µ
01014:	CD	205	=
01015:	10	016	▶

0100:0014

```
MOV SI, 00000h
MOV AL, [SI] + 02h
CMP AL, 00h
JZ 019h
MOV AH, 0Eh
INT 010h
INC SI
JMP 0Bh
MOV AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Ekran Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	0E	48
BX	00	00
CX	00	00
DX	00	00
CS	F400	
IP	0190	
SS	0100	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

F400:0190

F4190:	FF	255	RES
F4191:	FF	255	RES
F4192:	CD	205	=
F4193:	10	016	>
F4194:	CF	207	±
F4195:	00	000	NULL
F4196:	00	000	NULL
F4197:	00	000	NULL
F4198:	00	000	NULL
F4199:	00	000	NULL
F419A:	00	000	NULL
F419B:	00	000	NULL
F419C:	00	000	NULL
F419D:	00	000	NULL
F419E:	00	000	NULL
F419F:	00	000	NULL
F41A0:	FF	255	RES
F41A1:	FF	255	RES
F41A2:	CD	205	=
F41A3:	12	018	↓
F41A4:	CF	207	±
F41A5:	00	000	NULL

F400:0190

BIOS DI
INT 010h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADC CL, BH
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD BH, BH
DEC BP
ADC CX, DI
ADD [BX + SI], AL
ADD [BX + SI], AL
...

screen source reset aux vars debug stack fla

emulator screen (80x25 chars)

```
Hello
```

clear screen change font 0/16



# Ekranaya Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	0E	6F
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0017	
SS	0100	
SP	FFFE	
BP	0000	
SI	0005	
DI	0000	
DS	0100	
ES	0100	

0100:0017

```
01016: 46 070 F
01017: EB 235 6
01018: F2 242 2
01019: B4 180 1
0101A: 00 000 NULL
0101B: CD 205 =
0101C: 16 022 =
0101D: 90 144 É
0101E: 90 144 É
0101F: 90 144 É
01020: 90 144 É
01021: 90 144 É
01022: 90 144 É
01023: 90 144 É
01024: 90 144 É
01025: 90 144 É
01026: 90 144 É
01027: 90 144 É
01028: 90 144 É
01029: 90 144 É
0102A: 90 144 É
0102B: 90 144 É
```

0100:0017

```
INC SI
JMP 010Bh
MOV AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Ekрана Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	0E	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	000E	
SS	0100	
SP	FFFE	
BP	0000	
SI	0005	
DI	0000	
DS	0100	
ES	0100	

0100:000E

```
0100B: 8A 138 è
0100C: 44 068 D
0100D: 02 002 2
0100E: 3C 060 <
0100F: 00 000 NULL
01010: 74 116 t
01011: 07 007 BEEP
01012: B4 180 |
01013: 0E 014 |
01014: CD 205 =
01015: 10 016 >
01016: 46 070 F
01017: EB 235 6
01018: F2 242 >
01019: B4 180 |
0101A: 00 000 NULL
0101B: CD 205 =
0101C: 16 022 |
0101D: 90 144 é
0101E: 90 144 é
0101F: 90 144 é
01020: 90 144 é
```

0100:000E

```
MOV AL, [SI] + 02h
CMP AL, 00h
JZ 019h
MOV AH, 0Eh
INT 010h
INC SI
JMP 0Bh
MOV AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags





# Ekran Hello Yazdırma

emulator: hello.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	0E	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0019	
SS	0100	
SP	FFFE	
BP	0000	
SI	0005	
DI	0000	
DS	0100	
ES	0100	

0100:0019

0100B: 8A 138 è
0100C: 44 068 D
0100D: 02 002 0
0100E: 3C 060 <
0100F: 00 000 NULL
01010: 74 116 t
01011: 07 007 BEEP
01012: B4 180
01013: 0E 014 ʃ
01014: CD 205 =
01015: 10 016 ʃ
01016: 46 070 F
01017: EB 235 6
01018: F2 242 2
01019: B4 180
0101A: 00 000 NULL
0101B: CD 205 =
0101C: 16 022 ʃ
0101D: 90 144 é
0101E: 90 144 é
0101F: 90 144 é
01020: 90 144 é

0100:0019

```
MOV AL, [SI] + 02h
CMP AL, 00h
JZ 019h
MOV AH, 0Eh
INT 010h
INC SI
JMP 0Bh
MOV AH, 00h
INT 016h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# LED Ekran Testi

```
#start=led_display.exe#  
#make_bin#  
mov ax, 1234  
out 199, ax  
mov ax, -5678  
out 199, ax  
mov ax, 0  
x1:  
    out 199, ax  
    inc ax  
jmp x1  
hlt
```



emulator: led.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0000	0100:0000
01000: B8 184 7	MOU AX, 004D2h
01001: D2 210 0	OUT 0C7h, AX
01002: 04 004 4	MOU AX, 0E9D2h
01003: E7 231 7	OUT 0C7h, AX
01004: C7 199 0	MOU AX, 00000h
01005: B8 184 7	OUT 0C7h, AX
01006: D2 210 0	INC AX
01007: E9 233 0	JMP 0Dh
01008: E7 231 7	HLL
01009: C7 199 0	NOP
0100A: B8 184 7	NOP
0100B: 00 000 NULL	NOP
0100C: 00 000 NULL	NOP
0100D: E7 231 7	NOP
0100E: C7 199 0	NOP
0100F: 40 064 0	NOP
01010: EB 235 6	NOP
01011: FB 251 7	NOP
01012: F4 244 7	NOP
01013: 90 144 E	NOP
01014: 90 144 E	NOP
01015: 90 144 E	...

screen source reset aux vars debug stack flags





emulator: led.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	E9	D2
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	000A	
SS	0100	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

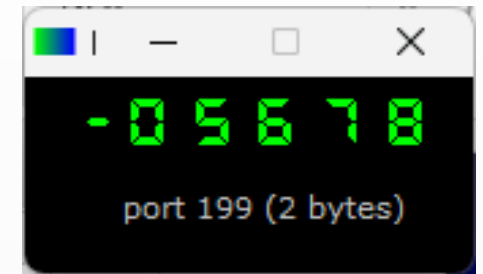
0100:000A

Address	Hex	Dec	Op
01000:	B8	184	⌵
01001:	D2	210	π
01002:	04	004	◆
01003:	E7	231	τ
01004:	C7	199	
01005:	B8	184	⌵
01006:	D2	210	π
01007:	E9	233	θ
01008:	E7	231	τ
01009:	C7	199	
0100A:	B8	184	⌵
0100B:	00	000	NULL
0100C:	00	000	NULL
0100D:	E7	231	τ
0100E:	C7	199	
0100F:	40	064	Ⓢ
01010:	EB	235	δ
01011:	FB	251	√
01012:	F4	244	∫
01013:	90	144	É
01014:	90	144	É
01015:	90	144	É

0100:000A

```
MOV AX, 004D2h
OUT 0C7h, AX
MOV AX, 0E9D2h
OUT 0C7h, AX
MOV AX, 00000h
OUT 0C7h, AX
INC AX
JMP 0Dh
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags





emulator: led.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	000F	
SS	0100	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:000F

Address	Hex	Dec	Symbol
01000:	B8	184	?
01001:	D2	210	π
01002:	04	004	◆
01003:	E7	231	τ
01004:	C7	199	
01005:	B8	184	?
01006:	D2	210	π
01007:	E9	233	θ
01008:	E7	231	τ
01009:	C7	199	
0100A:	B8	184	?
0100B:	00	000	NULL
0100C:	00	000	NULL
0100D:	E7	231	τ
0100E:	C7	199	
0100F:	40	064	0
01010:	EB	235	δ
01011:	FB	251	√
01012:	F4	244	∫
01013:	90	144	É
01014:	90	144	É
01015:	90	144	É

0100:000F

```
MOV AX, 004D2h
OUT 0C7h, AX
MOV AX, 0E9D2h
OUT 0C7h, AX
MOV AX, 00000h
OUT 0C7h, AX
INC AX
JMP 0Dh
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags

A small window with a black background and green digital characters. The top row shows five '0' characters. Below them, the text 'port 199 (2 bytes)' is displayed in white.



emulator: led.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	01
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0010	
SS	0100	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0010

01000:	B8	184	?
01001:	D2	210	π
01002:	04	004	◆
01003:	E7	231	τ
01004:	C7	199	
01005:	B8	184	?
01006:	D2	210	π
01007:	E9	233	θ
01008:	E7	231	τ
01009:	C7	199	
0100A:	B8	184	?
0100B:	00	000	NULL
0100C:	00	000	NULL
0100D:	E7	231	τ
0100E:	C7	199	
0100F:	40	064	0
01010:	EB	235	δ
01011:	FB	251	√
01012:	F4	244	∫
01013:	90	144	É
01014:	90	144	É
01015:	90	144	É

0100:0010

```
MOV AX, 004D2h
OUT 0C7h, AX
MOV AX, 0E9D2h
OUT 0C7h, AX
MOV AX, 00000h
OUT 0C7h, AX
INC AX
JMP 0Dh
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



emulator: led.bin\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	01
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0000

01000:	B8	184	⌘
01001:	D2	210	π
01002:	04	004	◆
01003:	E7	231	⌘
01004:	C7	199	
01005:	B8	184	⌘
01006:	D2	210	π
01007:	E9	233	⊖
01008:	E7	231	⌘
01009:	C7	199	
0100A:	B8	184	⌘
0100B:	00	000	NULL
0100C:	00	000	NULL
0100D:	E7	231	⌘
0100E:	C7	199	
0100F:	40	064	⓪
01010:	EB	235	δ
01011:	FB	251	√
01012:	F4	244	∫
01013:	90	144	É
01014:	90	144	É
01015:	90	144	É

0100:0000

```
OUT 0C7h, AX
INC AX
JMP 0Dh
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags







# Palindrom Kontrolü

```
jmp start
```

```
msg:
```

```
    str db '123321'
```

```
    str_size = $ - msg
```

```
    db 0Dh,0Ah,'$'
```

```
start:
```

```
    ; str ile belirtilen karakter dizisini ekrana yazdır
```

```
    mov ah, 9
```

```
    mov dx, offset str
```

```
    int 21h
```



# Palindrom Kontrolü

```
lea di, str ; str'nin adresini di yazmacına ata
mov si, di ; di dizinin başından başlar
add si, str_size ; si dizinin sonundan başlar
dec si ; dizideki son karakteri işaret et
mov cx, str_size
cmp cx, 1 ; karakter sayısını kontrol et
je is_palindrome ; tek karakter ise palindromdur
shr cx, 1 ; dizi boyunun yarısı kadar karşılaştırma gerekli
```



# Palindrom Kontrolü

next\_char:

```
mov al, [di]
```

```
mov bl, [si]
```

```
cmp al, bl
```

```
jne not_palindrome
```

```
inc di ; dizinin başından 1 ileri git
```

```
dec si ; dizinin sonundan 1 geri gel
```

```
loop next_char
```



# Palindrom Kontrolü

is\_palindrome:

; verilen dizi palindrom, ekrana yazdır

mov ah, 9

mov dx, offset msg1

int 21h

jmp stop



# Palindrom Kontrolü

not\_palindrome:

; palindrom değil yazdır

mov ah, 9

mov dx, offset msg2

int 21h

stop:

msg1 db "palindrome!\$"

msg2 db "not palindrome!\$"



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	60
DX	00	00
CS	07 00	
IP	01 0B	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0B

```
07100: EB 235 6
07101: 09 009 TAB
07102: 31 049 1
07103: 32 050 2
07104: 33 051 3
07105: 33 051 3
07106: 32 050 2
07107: 31 049 1
07108: 0D 013 CRET
07109: 0A 010 NEWL
0710A: 24 036 $
0710B: B4 180 |
0710C: 09 009 TAB
0710D: BA 186 ||
0710E: 02 002 @
0710F: 01 001 @
07110: CD 205 =
07111: 21 033 !
07112: BF 191 7
07113: 02 002 @
07114: 01 001 @
07115: 8B 139 i
```

07 00: 01 0B

```
MOU AH, 09h
MOU DX, 00102h
INT 021h
MOU DI, 00102h
MOU SI, DI
ADD SI, 06h
DEC SI
MOU CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOU AL, [DI]
MOU BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOU AH, 09h
MOU DX, 00142h
INT 021h
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	09	00
BX	00	00
CX	00	60
DX	00	00
CS	07 00	
IP	01 0D	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0D

```
07100: EB 235 6
07101: 09 009 TAB
07102: 31 049 1
07103: 32 050 2
07104: 33 051 3
07105: 33 051 3
07106: 32 050 2
07107: 31 049 1
07108: 0D 013 CRET
07109: 0A 010 NEWL
0710A: 24 036 $
0710B: B4 180 |
0710C: 09 009 TAB
0710D: BA 186 ||
0710E: 02 002 @
0710F: 01 001 @
07110: CD 205 =
07111: 21 033 !
07112: BF 191 7
07113: 02 002 @
07114: 01 001 @
07115: 8B 139 i
```

07 00: 01 0D

```
MOU AH, 09h
MOU DX, 00102h
INT 021h
MOU DI, 00102h
MOU SI, DI
ADD SI, 06h
DEC SI
MOU CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOU AL, [DI]
MOU BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOU AH, 09h
MOU DX, 00142h
INT 021h
...
```

screen source reset aux vars debug stack flags





# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	00
BX	00	00
CX	00	60
DX	01	02
CS	0700	
IP	0110	
SS	0700	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0700:0110			0700:0110		
07100:	EB	235	δ	MOU	AH, 09h
07101:	09	009	TAB	MOU	DX, 00102h
07102:	31	049	1	INT	021h
07103:	32	050	2	MOU	DI, 00102h
07104:	33	051	3	MOU	SI, DI
07105:	33	051	3	ADD	SI, 06h
07106:	32	050	2	DEC	SI
07107:	31	049	1	MOU	CX, 00006h
07108:	0D	013	CRET	CMP	CX, 01h
07109:	0A	010	NEWL	JZ	0131h
0710A:	24	036	\$	SHR	CX, 1
0710B:	B4	180		MOU	AL, [DI]
0710C:	09	009	TAB	MOU	BL, [SI]
0710D:	BA	186		CMP	AL, BL
0710E:	02	002	@	JNE	013Ah
0710F:	01	001	@	INC	DI
07110:	CD	205	=	DEC	SI
07111:	21	033	!	LOOP	0125h
07112:	BF	191	1	MOU	AH, 09h
07113:	02	002	@	MOU	DX, 00142h
07114:	01	001	@	INT	021h
07115:	8B	139	i	...	

screen source reset aux vars debug stack fla

emulator screen (80x25 chars)

```
123321
```

clear screen change font 0/16



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	24
BX	00	00
CX	00	60
DX	01	02
CS	07 00	
IP	0112	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	07 00	
ES	07 00	

```
0700:0112 07112: BF 191 7
07113: 02 002 0
07114: 01 001 0
07115: 8B 139 i
07116: F7 247 ≈
07117: 83 131 â
07118: C6 198 †
07119: 06 006 †
0711A: 4E 078 N
0711B: B9 185 †
0711C: 06 006 †
0711D: 00 000 NULL
0711E: 83 131 â
0711F: F9 249 ·
07120: 01 001 0
07121: 74 116 t
07122: 0E 014 †
07123: D1 209 †
07124: E9 233 0
07125: 8A 138 è
07126: 05 005 †
07127: 8A 138 è
```

```
MOU DI, 00102h
MOU SI, DI
ADD SI, 06h
DEC SI
MOU CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOU AL, [DI]
MOU BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOU AH, 09h
MOU DX, 00142h
INT 021h
JMP 0141h
MOU AH, 09h
MOU DX, 0014Eh
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	24
BX	00	00
CX	00	60
DX	01	02
CS	07 00	
IP	0115	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0000	
DI	0102	
DS	07 00	
ES	07 00	

```
07112: BF 191 7
07113: 02 002 0
07114: 01 001 0
07115: 8B 139 i
07116: F7 247 ≈
07117: 83 131 à
07118: C6 198 †
07119: 06 006 †
0711A: 4E 078 N
0711B: B9 185 †
0711C: 06 006 †
0711D: 00 000 NULL
0711E: 83 131 à
0711F: F9 249 ·
07120: 01 001 0
07121: 74 116 t
07122: 0E 014 †
07123: D1 209 †
07124: E9 233 0
07125: 8A 138 è
07126: 05 005 †
07127: 8A 138 è
```

```
MOU DI, 00102h
MOU SI, DI
ADD SI, 06h
DEC SI
MOU CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOU AL, [DI]
MOU BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOU AH, 09h
MOU DX, 00142h
INT 021h
JMP 0141h
MOU AH, 09h
MOU DX, 0014Eh
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	24
BX	00	00
CX	00	60
DX	01	02
CS	07 00	
IP	0117	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0102	
DI	0102	
DS	07 00	
ES	07 00	

```
07112: BF 191 7
07113: 02 002 0
07114: 01 001 0
07115: 8B 139 i
07116: F7 247 ≈
07117: 83 131 à
07118: C6 198 †
07119: 06 006 †
0711A: 4E 078 N
0711B: B9 185 il
0711C: 06 006 †
0711D: 00 000 NULL
0711E: 83 131 à
0711F: F9 249 ·
07120: 01 001 0
07121: 74 116 t
07122: 0E 014 ʘ
07123: D1 209 ʘ
07124: E9 233 0
07125: 8A 138 è
07126: 05 005 †
07127: 8A 138 è
```

```
MOV DI, 00102h
MOV SI, DI
ADD SI, 06h
DEC SI
MOV CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOV AL, [DI]
MOV BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOV AH, 09h
MOV DX, 00142h
INT 021h
JMP 0141h
MOV AH, 09h
MOV DX, 0014Eh
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	24
BX	00	00
CX	00	60
DX	01	02
CS	07 00	
IP	011A	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0108	
DI	0102	
DS	07 00	
ES	07 00	

```
07112: BF 191 7
07113: 02 002 0
07114: 01 001 0
07115: 8B 139 i
07116: F7 247 ≈
07117: 83 131 â
07118: C6 198 †
07119: 06 006 †
0711A: 4E 078 N
0711B: B9 185 †
0711C: 06 006 †
0711D: 00 000 NULL
0711E: 83 131 â
0711F: F9 249 ·
07120: 01 001 0
07121: 74 116 t
07122: 0E 014 ʘ
07123: D1 209 ʘ
07124: E9 233 0
07125: 8A 138 è
07126: 05 005 †
07127: 8A 138 è
```

```
MOV DI, 00102h
MOV SI, DI
ADD SI, 06h
DEC SI
MOV CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOV AL, [DI]
MOV BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOV AH, 09h
MOV DX, 00142h
INT 021h
JMP 0141h
MOV AH, 09h
MOV DX, 0014Eh
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	24
BX	00	00
CX	00	06
DX	01	02
CS	07 00	
IP	0123	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0107	
DI	0102	
DS	07 00	
ES	07 00	

0700:0123

```
07112: BF 191 7
07113: 02 002 0
07114: 01 001 0
07115: 8B 139 i
07116: F7 247 ≈
07117: 83 131 â
07118: C6 198 †
07119: 06 006 †
0711A: 4E 078 N
0711B: B9 185 †
0711C: 06 006 †
0711D: 00 000 NULL
0711E: 83 131 â
0711F: F9 249 ·
07120: 01 001 0
07121: 74 116 t
07122: 0E 014 †
07123: D1 209 †
07124: E9 233 0
07125: 8A 138 è
07126: 05 005 †
07127: 8A 138 è
```

0700:0123

```
MOV DI, 00102h
MOV SI, DI
ADD SI, 06h
DEC SI
MOV CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOV AL, [DI]
MOV BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOV AH, 09h
MOV DX, 00142h
INT 021h
JMP 0141h
MOV AH, 09h
MOV DX, 0014Eh
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	24
BX	00	00
CX	00	03
DX	01	02
CS	07 00	
IP	0125	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0107	
DI	0102	
DS	07 00	
ES	07 00	

```
0700:0125 07112: BF 191 7
07113: 02 002 0
07114: 01 001 0
07115: 8B 139 i
07116: F7 247 ≈
07117: 83 131 â
07118: C6 198 †
07119: 06 006 †
0711A: 4E 078 N
0711B: B9 185 il
0711C: 06 006 †
0711D: 00 000 NULL
0711E: 83 131 â
0711F: F9 249 ·
07120: 01 001 0
07121: 74 116 t
07122: 0E 014 ʘ
07123: D1 209 ʘ
07124: E9 233 0
07125: 8A 138 è
07126: 05 005 †
07127: 8A 138 è
```

```
MOV DI, 00102h
MOV SI, DI
ADD SI, 06h
DEC SI
MOV CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOV AL, [DI]
MOV BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOV AH, 09h
MOV DX, 00142h
INT 021h
JMP 0141h
MOV AH, 09h
MOV DX, 0014Eh
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	31
BX	00	00
CX	00	03
DX	01	02
CS	07 00	
IP	0127	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0107	
DI	0102	
DS	07 00	
ES	07 00	

```
0700:0127 07112: BF 191 7
07113: 02 002 0
07114: 01 001 0
07115: 8B 139 i
07116: F7 247 ≈
07117: 83 131 â
07118: C6 198 †
07119: 06 006 †
0711A: 4E 078 N
0711B: B9 185 il
0711C: 06 006 †
0711D: 00 000 NULL
0711E: 83 131 â
0711F: F9 249 ·
07120: 01 001 0
07121: 74 116 t
07122: 0E 014 ʘ
07123: D1 209 ʘ
07124: E9 233 0
07125: 8A 138 è
07126: 05 005 †
07127: 8A 138 è

MOV DI, 00102h
MOV SI, DI
ADD SI, 06h
DEC SI
MOV CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOV AL, [DI]
MOV BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOV AH, 09h
MOV DX, 00142h
INT 021h
JMP 0141h
MOV AH, 09h
MOV DX, 0014Eh
...
```

screen source reset aux vars debug stack flags





# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	31
BX	00	31
CX	00	03
DX	01	02
CS	07 00	
IP	0129	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0107	
DI	0102	
DS	07 00	
ES	07 00	

```
07129: 3A 058 :
0712A: C3 195 |
0712B: 75 117 u
0712C: 0D 013 CRET
0712D: 47 071 G
0712E: 4E 078 N
0712F: E2 226 r
07130: F4 244 f
07131: B4 180 |
07132: 09 009 TAB
07133: BA 186 ||
07134: 42 066 B
07135: 01 001 @
07136: CD 205 =
07137: 21 033 ?
07138: EB 235 d
07139: 07 007 BEEP
0713A: B4 180 |
0713B: 09 009 TAB
0713C: BA 186 ||
0713D: 4E 078 N
0713E: 01 001 @
```

```
MOU DI, 00102h
MOU SI, DI
ADD SI, 06h
DEC SI
MOU CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOU AL, [DI]
MOU BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOU AH, 09h
MOU DX, 00142h
INT 021h
JMP 0141h
MOU AH, 09h
MOU DX, 0014Eh
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	09	31
BX	00	31
CX	00	03
DX	01	02
CS	07 00	
IP	012B	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0107	
DI	0102	
DS	07 00	
ES	07 00	

0700:012B

```
07129: 3A 058 :  
0712A: C3 195 |  
0712B: 75 117 u  
0712C: 0D 013 CRET  
0712D: 47 071 G  
0712E: 4E 078 N  
0712F: E2 226 r  
07130: F4 244 |  
07131: B4 180 |  
07132: 09 009 TAB  
07133: BA 186 ||  
07134: 42 066 B  
07135: 01 001 @  
07136: CD 205 =  
07137: 21 033 ?  
07138: EB 235 d  
07139: 07 007 BEEP  
0713A: B4 180 |  
0713B: 09 009 TAB  
0713C: BA 186 ||  
0713D: 4E 078 N  
0713E: 01 001 @
```

0700:012B

```
MOV DI, 00102h  
MOV SI, DI  
ADD SI, 06h  
DEC SI  
MOV CX, 00006h  
CMP CX, 01h  
JZ 0131h  
SHR CX, 1  
MOV AL, [DI]  
MOV BL, [SI]  
CMP AL, BL  
JNE 013Ah  
INC DI  
DEC SI  
LOOP 0125h  
MOV AH, 09h  
MOV DX, 00142h  
INT 021h  
JMP 0141h  
MOV AH, 09h  
MOV DX, 0014Eh  
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	31
BX	00	31
CX	00	03
DX	01	02
CS	07 00	
IP	012D	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0107	
DI	0102	
DS	07 00	
ES	07 00	

```
07129: 3A 058 :
0712A: C3 195 |
0712B: 75 117 u
0712C: 0D 013 CRET
0712D: 47 071 G
0712E: 4E 078 N
0712F: E2 226 r
07130: F4 244 |
07131: B4 180 |
07132: 09 009 TAB
07133: BA 186 ||
07134: 42 066 B
07135: 01 001 @
07136: CD 205 =
07137: 21 033 ?
07138: EB 235 d
07139: 07 007 BEEP
0713A: B4 180 |
0713B: 09 009 TAB
0713C: BA 186 ||
0713D: 4E 078 N
0713E: 01 001 @
```

```
MOV DI, 00102h
MOV SI, DI
ADD SI, 06h
DEC SI
MOV CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOV AL, [DI]
MOV BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOV AH, 09h
MOV DX, 00142h
INT 021h
JMP 0141h
MOV AH, 09h
MOV DX, 0014Eh
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	31
BX	00	31
CX	00	03
DX	01	02
CS	07 00	
IP	012E	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0107	
DI	0103	
DS	07 00	
ES	07 00	

```
07129: 3A 058 :
0712A: C3 195 |
0712B: 75 117 u
0712C: 0D 013 CRET
0712D: 47 071 G
0712E: 4E 078 N
0712F: E2 226 r
07130: F4 244 |
07131: B4 180 |
07132: 09 009 TAB
07133: BA 186 ||
07134: 42 066 B
07135: 01 001 @
07136: CD 205 =
07137: 21 033 ?
07138: EB 235 d
07139: 07 007 BEEP
0713A: B4 180 |
0713B: 09 009 TAB
0713C: BA 186 ||
0713D: 4E 078 N
0713E: 01 001 @
```

```
MOV DI, 00102h
MOV SI, DI
ADD SI, 06h
DEC SI
MOV CX, 00006h
CMP CX, 01h
JZ 0131h
SHR CX, 1
MOV AL, [DI]
MOV BL, [SI]
CMP AL, BL
JNE 013Ah
INC DI
DEC SI
LOOP 0125h
MOV AH, 09h
MOV DX, 00142h
INT 021h
JMP 0141h
MOV AH, 09h
MOV DX, 0014Eh
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	09	31
BX	00	31
CX	00	03
DX	01	02
CS	07 00	
IP	012F	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0106	
DI	0103	
DS	07 00	
ES	07 00	

0700:012F

```
07129: 3A 058 :  
0712A: C3 195 |  
0712B: 75 117 u  
0712C: 0D 013 CRET  
0712D: 47 071 G  
0712E: 4E 078 N  
0712F: E2 226 r  
07130: F4 244 r  
07131: B4 180 |  
07132: 09 009 TAB  
07133: BA 186 ||  
07134: 42 066 B  
07135: 01 001 @  
07136: CD 205 =  
07137: 21 033 ?  
07138: EB 235 d  
07139: 07 007 BEEP  
0713A: B4 180 |  
0713B: 09 009 TAB  
0713C: BA 186 ||  
0713D: 4E 078 N  
0713E: 01 001 @
```

0700:012F

```
MOV DI, 00102h  
MOV SI, DI  
ADD SI, 06h  
DEC SI  
MOV CX, 00006h  
CMP CX, 01h  
JZ 0131h  
SHR CX, 1  
MOV AL, [DI]  
MOV BL, [SI]  
CMP AL, BL  
JNE 013Ah  
INC DI  
DEC SI  
LOOP 0125h  
MOV AH, 09h  
MOV DX, 00142h  
INT 021h  
JMP 0141h  
MOV AH, 09h  
MOV DX, 0014Eh  
...
```

screen source reset aux vars debug stack flags



# Palindrom Kontrolü

emulator: pali.com\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers	H	L
AX	09	33
BX	00	33
CX	00	00
DX	01	42
CS	0700	
IP	0136	
SS	0700	
SP	FFFE	
BP	0000	
SI	0104	
DI	0105	
DS	0700	
ES	0700	

0700:0136		0700:0136	
07125:	8A 138	è	MOU DI, 00102h
07126:	05 005	⌘	MOU SI, DI
07127:	8A 138	è	ADD SI, 06h
07128:	1C 028	:	DEC SI
07129:	3A 058	:	MOU CX, 00006h
0712A:	C3 195		CMP CX, 01h
0712B:	75 117	u	JZ 0131h
0712C:	0D 013	CRET	SHR CX, 1
0712D:	47 071	G	MOU AL, [DI]
0712E:	4E 078	N	MOU BL, [SI]
0712F:	E2 226	Γ	CMP AL, BL
07130:	F4 244	↑	JNE 013Ah
07131:	B4 180		INC DI
07132:	09 009	TAB	DEC SI
07133:	BA 186		LOOP 0125h
07134:	42 066	B	MOU AH, 09h
07135:	01 001	©	MOU DX, 00142h
07136:	CD 205	=	INT 021h
07137:	21 033	!	JMP 0141h
07138:	EB 235	δ	MOU AH, 09h
07139:	07 007	BEEP	MOU DX, 0014Eh
0713A:	B4 180		...

screen source reset aux vars debug stack fla

emulator screen (80x25 chars)

```
123321
palindrome!
```

clear screen change font 0/16



SON