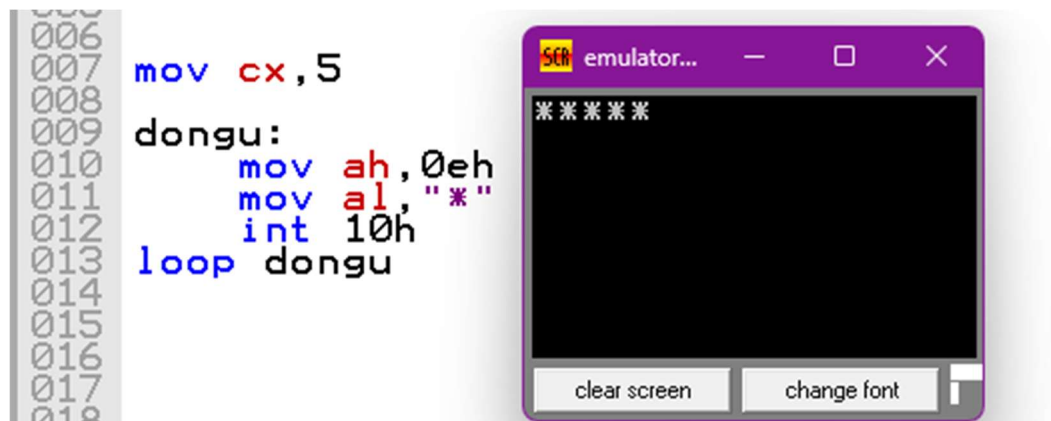
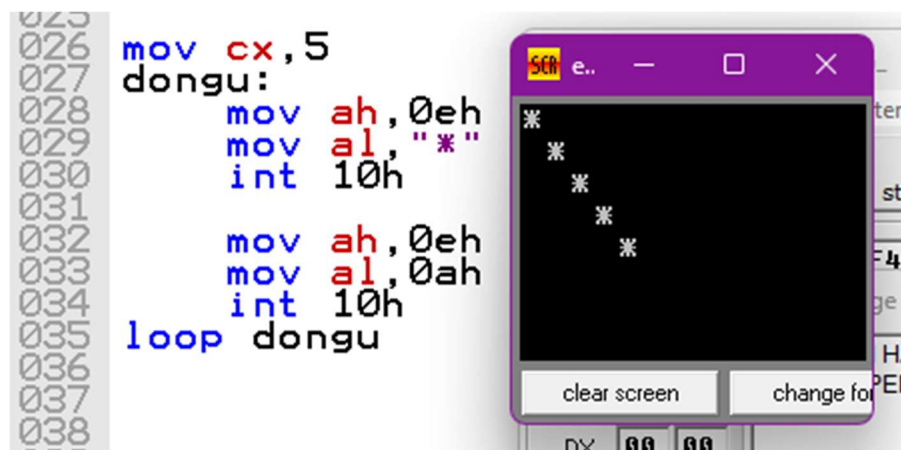


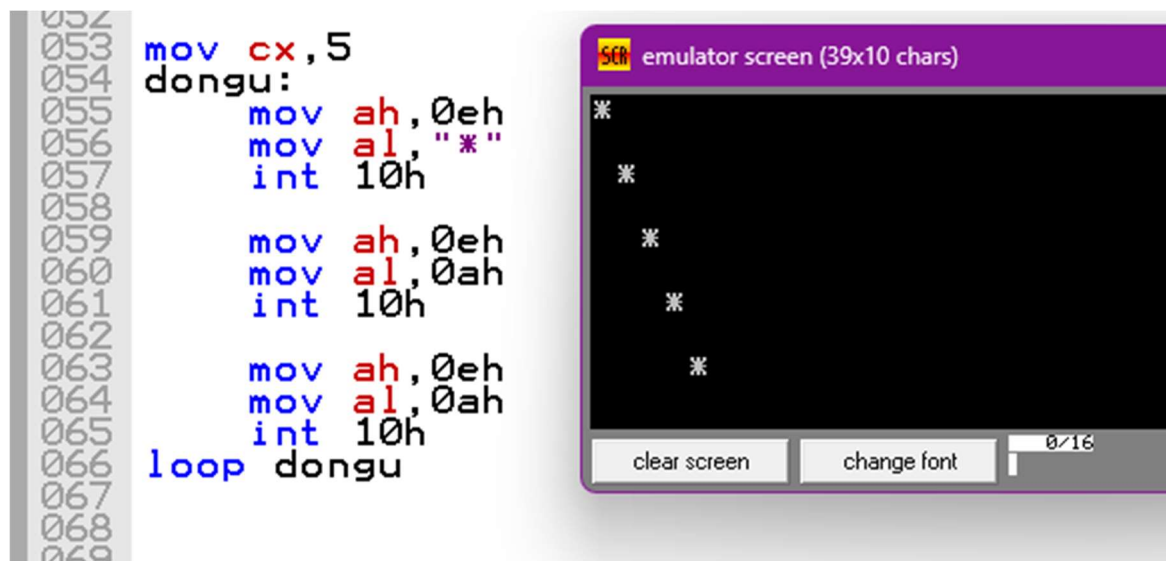
**SORU 1- \*\*\*\*\* SEKLINI CIZEN KODU YAZINIZ**



**SORU 2- ASAGIDAKI SEKLI CIZEN KODU YAZINIZ**

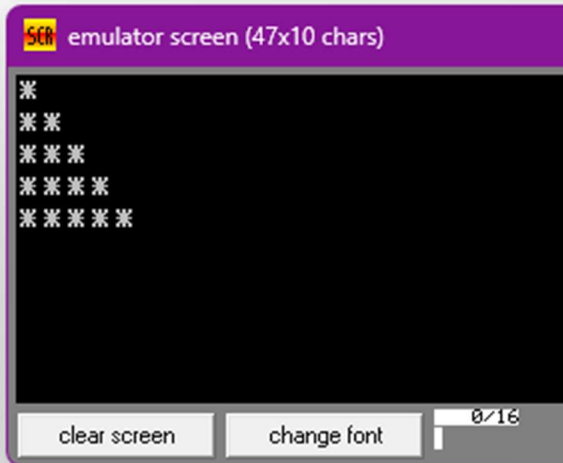


**SORU 3- ASAGIDAKI SEKLI CIZEN KODU YAZINIZ**




**SORU 4- AŞAĞIDAKİ ŞEKLİ ÇİZEN KODU YAZINIZ**

```
074  
075 mov cx,1  
076  
077 dongu:  
078     mov al,"*"  
079     mov ah,0ah  
080     int 10h  
081  
082     mov ah,0eh  
083     mov al,0ah  
084     int 10h  
085  
086     inc cx  
087     cmp cx,6  
088     je son  
089     jmp dongu  
090  
091 son:hlt  
092  
093  
094
```



**SORU 5- AŞAĞIDAKİ ŞEKLİ ÇİZEN KODU YAZINIZ.**

```
01 org 100h  
02  
03 mov cx,5  
04  
05 dongu:  
06     mov al,"*"  
07     mov ah,0ah  
08     int 10h  
09  
10     mov ah,0eh  
11     mov al,0ah  
12     int 10h  
13  
14  
15 loop dongu  
16  
17  
18 ret
```



**SORU 6 - AŞAĞIDAKİ ŞEKLİ ÇİZEN KODU YAZINIZ.**


```
01 org 100h
02 mov cx, 8
03 mov ah, 0eh
04
05 anaDongu:
06     push cx
07     icDongu:
08         mov al, '*'
09         int 10h
10     loop icDongu
11
12     mov al, ' '
13     int 10h
14     pop cx
15 loop anaDongu
16 hlt
```

**SCR** emulator screen (80x25 chars)

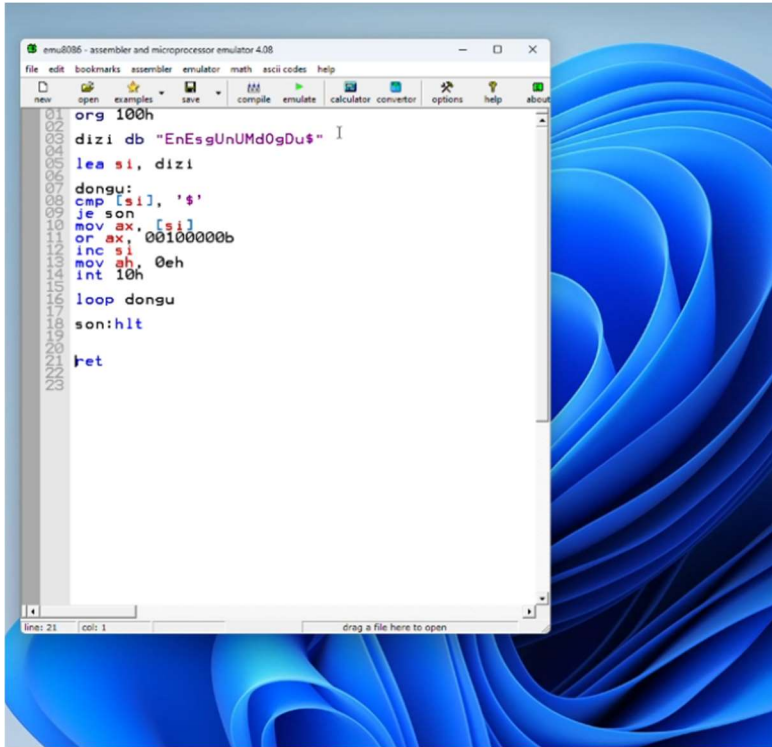
\*\*\*\*\*

**SORU – 7 BÜYÜK HARFLERİ KÜÇÜK HARF YAPAN KODU YAZINIZ**

```
new open examples save compile emulate calculator convertor option:
01 org 100h
02
03
04 lea si,dizi
05
06
07 dongu:
08     cmp [si],'$'
09     je son
10     mov ax,[si]
11     or ax,00100000b
12     inc si
13     mov ah,0eh
14     int 10h
15
16     loop dongu
17
18 son:
19 hlt
20
21
22
23 ret
24
25 dizi db "DeNeMe$"
26
27
28
```

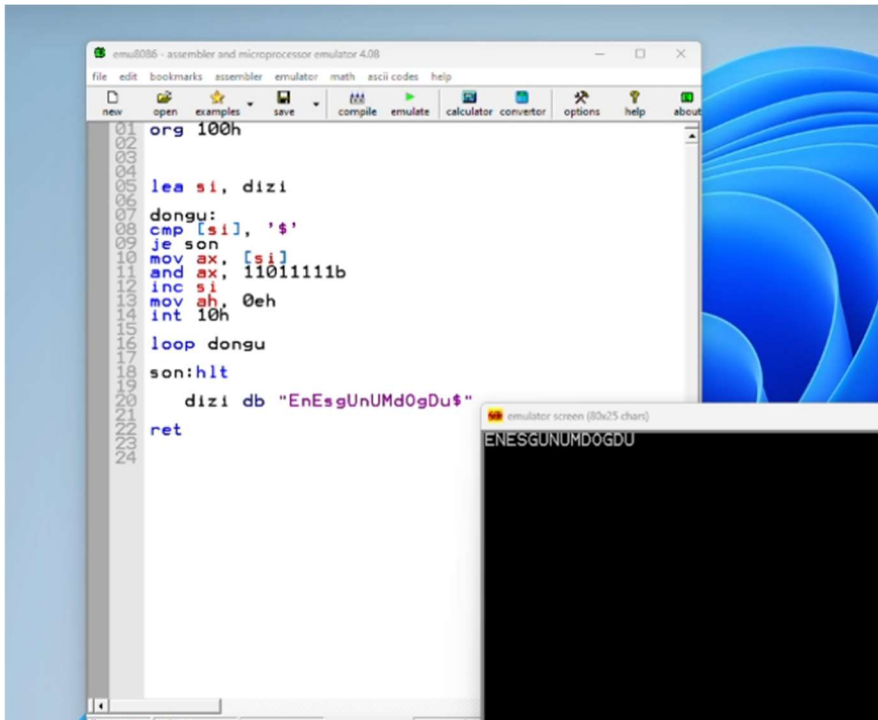


## SORU 8 – DİZİDEKİ TÜM HARFLERİ KÜÇÜLTEN KODU YAZINIZ



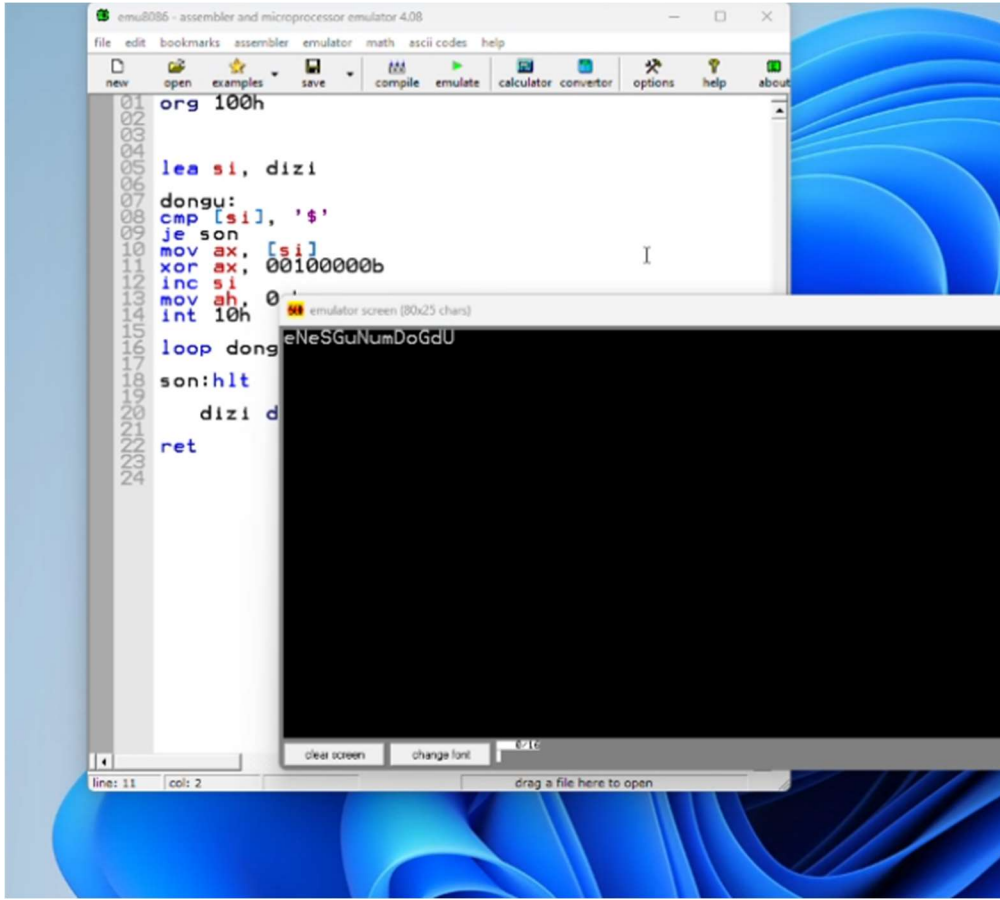
```
org 100h
dizi db "EnEsgUnUMd0gDu$"
lea si, dizi
dongu:
cmp [si], '$'
je son
mov ax, [si]
or ax, 00100000b
inc si
mov ah, 0eh
int 10h
loop dongu
son: hlt
ret
```

## SORU 9 – DİZİDEKİ TÜM HARFLERİ BÜYÜTEN KODU YAZINIZ



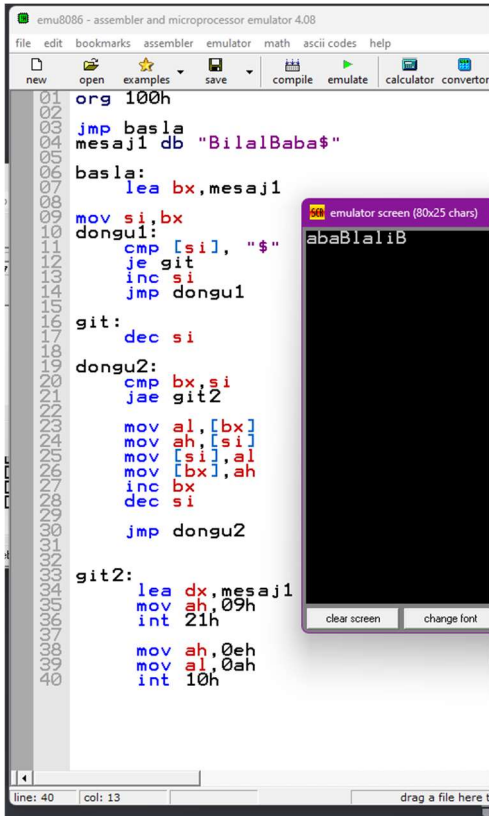
```
org 100h
lea si, dizi
dongu:
cmp [si], '$'
je son
mov ax, [si]
and ax, 11011111b
inc si
mov ah, 0eh
int 10h
loop dongu
son: hlt
dizi db "EnEsgUnUMd0gDu$"
ret
```

## SORU 10 – BÜYÜKLERİ KÜÇÜK KÜÇÜKLERİ BÜYÜK YAPAN KODU YAZINIZ



```
01 org 100h
02
03
04
05 lea si, dizi
06
07 dongu:
08 cmp [si], '$'
09 je son
10 mov ax, [si]
11 xor ax, 00100000b
12 inc si
13 mov ah, 0
14 int 10h
15 loop dongu
16
17 son: hlt
18
19 dizi db "BilalBaba$"
20
21 ret
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
```

## SORU 11 – VERİLEN İFADEYİ DOLARI ALMADAN TERSTEN YAZAN KOD



```
01 org 100h
02
03 jmp basla
04 mesaj1 db "BilalBaba$"
05
06 basla:
07 lea bx, mesaj1
08 mov si, bx
09 dongu1:
10 cmp [si], "$"
11 je git
12 inc si
13 jmp dongu1
14
15 git:
16 dec si
17
18 dongu2:
19 cmp bx, si
20 jae git2
21
22 mov al, [bx]
23 mov ah, [si]
24 mov [si], al
25 mov [bx], ah
26 inc bx
27 dec si
28 jmp dongu2
29
30 git2:
31 lea dx, mesaj1
32 mov ah, 09h
33 int 21h
34
35 mov ah, 0eh
36 mov al, 0ah
37 int 10h
38
39
40
```

**SORU 12- KÜÇÜK OLAN SAYIYI VEREN KOD(2 PARAMETRELİ)**

```
ASMdenDegerAl proc
    mov eax, ecx
    cmp edx, eax
    jg Return
    mov eax, edx
Return:
ret
ASMdenDegerAl endp
```

**SORU 13 – FIBONACCI**

```
Fibonacci :
Fibonacci proc
    mov rax, 0
    mov rbx, 1
    mov rcx, 9
dongu:
    push ax
    add rax, rbx
    xchg rax, rbx
    loop dongu
fibonacci endp
end
```