

1.

```
org 0100h
mov [0100h], '1'
mov [0101h], '4'
mov [0102h], '2'
mov [0103h], '8'
mov [0104h], '7'
mov [0105h], '3'
mov [0106h], '5'
mov [0107h], '9'
mov [0108h], '2'
mov [0109h], '6'

call ekranayazdir    ; baslangic durumunu yazdir

mov bx,0100h        ; ilk sayi
mov cx,0109h        ; son sayi
mov dl,2            ; bolen

teksayidongu:        ; soldaki ilk cift sayiyi bulur
    mov si, bx
    mov al,[si]
    xor ah,ah        ; onceki bolme sonucu atiliyor
    div dl
    cmp ah,0         ; bolmede kalan varmi?
    jnz devamtek     ; kalan varsa tektir
    jmp ciftsayidongu ; kalan yoksa cifttir

    devamtek:
        inc bx
        cmp bx,cx
        je cikis
        jmp teksayidongu

ciftsayidongu:
    mov si,cx
    mov al,[si]
    xor ah,ah
    div dl
    cmp ah,0
    jz devamcift
    jmp degistir

    devamcift:
        dec cx
        cmp bx,cx
        je cikis
        jmp ciftsayidongu

degistir:
    push ax
    mov al,[bx]
    mov ah,[si]
    mov [bx],ah
    mov [si],al
    pop ax
    jmp teksayidongu
```

```

cikis:
call ekranayazdir
hlt

ekranayazdir proc
    mov cx, 0                ; karakter sayaci
    mov ah, 2                ; int 21, ah=2, ekrana karakter yazdir
    tekrar:
        mov si, cx
        mov dl, [0100h+si]
        int 21h              ; int 21, ah=2, ekrana karakter yazdir
        inc cx
        cmp cx, 9
        ja yazmacikisi
        mov dl, ','
        int 21h              ; int 21, ah=2, ekrana karakter yazdir
    jmp tekrar
yazmacikisi:
    mov dl, 0Dh              ; return
    int 21h
    mov dl, 0Ah              ; yeni satir
    int 21h
    ret
ekranayazdir endp

```

2. Aşağıdaki program iki sayının çarpımını toplama ve bölme ile yapmaktadır. Her döngüde birinci sayı kendisiyle toplanarak iki katına çıkarılmakta ikinci sayı ise 2'ye bölünerek yarıya düşürülmektedir. Böylece ikinci sayı 1 olduğunda birinci sayı sonuç olacaktır. Ancak tamsayı bölme yapıldığı için her adımdaki kalan değer sonuç değeri saklayan cx register'ına eklenmektedir. İkinci sayının 1 olduğu değerde birinci sayının değerinin eledilmesi için bolum sonucunun 0 olduğu (al=0) duruma kadar gidilmekte ve bu aşamada kalan değer (ah=1) için bx değeri cx sonucuna eklenmektedir.

Örnek hesaplama:

cx	bx	ax	kalan	açıklama
0	121	235	-	; başlangıç
121	242	117	1	; 1.adım 121 eklenir (kalan 1)
363	484	58	1	; 2.adım 242 eklenir (kalan 1)
363	968	29	0	; 3.adım ekleme yapılmaz (kalan 0)
1331	1936	14	1	; 4.adım 968 eklenir (kalan 1)
1331	3872	7	0	; 5.adım ekleme yapılmaz (kalan 0)
5203	7744	3	1	; 6.adım 3872 eklenir (kalan 1)
12947	15488	1	1	; 7.adım 7744 eklenir (kalan 1)
28435	-	0	1	; 8.adım 15488 eklenir (kalan 1)

```

org 0100h
mov [0100h], word ptr 121 ; 1.sayi
mov [0102h], word ptr 235 ; 2.sayi
mov bx, [0100h]
mov ax, [0102h]
mov dl, 2                ; bolen (2)
mov cx, 0                ; kalanların toplamı saklanacak
cmp bx, 0                ; 1.sayı sıfırmı?
jz bitis                 ; sıfırsa bitir

```

```

dongu:                ; her dongude 1.sayi iki katina cikar 2.sayi 2'ye bolunur
    cmp ax,0          ; 2.sayi sifirsa veya dongude sifir olduysa bitir
    jz bitis
    div dl             ; 2.sayi 2'ye bolunur
    cmp ah,0          ; kalan kontrol ediliyor
    jz atla           ; kalan yoksa ekleme yapmadan devam et
    add cx,bx          ; kalan 1 ise 1.sayiye sonuca ekle
atla:
    add bx,bx          ; 1.sayiye iki katina cikar
    xor ah,ah          ; bir oncedeki islemden kalani sifirla
    jmp dongu
bitis:
mov [0104h],cx        ; sonucu bir sonraki adrese kaydet
hlt

```

3.

```

org 0100h
mov [0100h],'7'
mov [0101h],'5'
mov [0102h],'9'
mov [0103h],'3'
mov [0104h],'2'
mov [0105h],'7'
mov [0106h],'8'
mov [0107h],'6'
mov [0108h],'5'
mov [0109h],'1'
mov [010Ah],'3'

call ekranayazdir    ; baslangic durumunu yazdir

mov dx,0100h         ; soldaki gosterge
mov cx,0100h         ; kontrol edilen
mov bx,0100h         ; en kucuk sayinin adresi icin

dongu:                ; her sayinin dogru yeri bulunur
    mov si, cx
    mov al,[si]
    cmp al,[bx]       ; bir sonraki sayi en kucukle karsilastirilir
    jae atla          ; buyuk veya esitse bir sonrakine gec
    mov bx,cx         ; yeni sayi en kucukten kucukse en kucuk yap
    inc cx            ; bir sonrakine gec
    cmp cx,010Bh      ; sona gelindimi?
    je yerdegistir    ; sonda ise yer degistir
    jmp dongu         ; sona gelinmedi basa don
yerdegistir:
    cmp bx,dx
    je ayniatla       ; ayni eleman yer degistirme
    call degistir     ; dx ile bx yer degistirir
ayniatla:
    inc dx            ; soldaki gosterge bir ilerler
    cmp dx, 010Ah     ; sona gelindimi?
    je cikis          ; son elemanda sirasina konduysa cikis yap
    mov cx,dx         ; siralanmamis sayinin adresi cx bx e atilir
    mov bx,dx
    jmp dongu         ; basa don
atla:                ; yeni sayi buyuk veya esitse bir sonrakini sec
    inc cx
    cmp cx,010Bh      ; sona gelindimi?

```

```

        je yerdegistir      ; son elemanda en kucukle karsilastirildigi
        jmp dongu           ; yer degirme yapilir ve basa donulur

cikis:
        call ekranayazdir
        hlt

degistir proc
        push ax
        push si
        mov si,dx
        mov al,[si]
        mov ah,[bx]
        mov [si],ah
        mov [bx],al
        pop si
        pop ax
        ret
degistir endp

ekranayazdir proc
        mov cx, 0           ; karakter sayaci
        mov ah, 2           ; int 21, ah=2, ekrana karakter yazdir
        tekrar:
            mov si,cx
            mov dl, [0100h+si]
            int 21h          ; int 21, ah=2, ekrana karakter yazdir
            inc cx
            cmp cx,11
            je yazmacikisi
            mov dl, ','
            int 21h          ; int 21, ah=2, ekrana karakter yazdir
        jmp tekrar
        yazmacikisi:
            mov dl, 0Dh      ; return
            int 21h
            mov dl, 0Ah      ; yeni satir
            int 21h
            ret
ekranayazdir endp

```

Ekstra

```

org 0100h
ADET equ 1000           ; toplam rakam sayisi
mov dl,10               ; sayilari onluk tabanda gosterme icin

call baslangic          ; baslangic degerleri atanir
call degerata           ; iki sayinin degerleri atanir
call toplama            ; toplama islemi yapilir
call mesajyaz           ; kullaniciya islemin bittigi bildirilir

        hlt

topla proc
        call baslangic
        mov bx,0         ; bl'de kalan saklanacak
        mov cx,ADET      ; cx sayaci rakam sayisina esitleniyor
        sonrakibasamak:
            mov al,[si]   ; ilk sayinin en sagdaki basamagi

```

```
        add al,[si+1000]      ; ikinci sayinin en sagdaki basamagiyla toplaniyor
        add al,bl             ; onceki basamaktan gelen toplandi
        div dl                ; toplam 10'a bolunerek kanal alinacak
        mov [si+2001],ah      ; elde olursa 1001 byte gerekir o yuzden 2001
        xor ah,ah             ; bir sonraki bolme icin
        mov bl,al             ; kalan saklandi sonraki basamaga eklenecek
        dec si                ; onceki basamaklara gecilir
        loop sonrakibasamak
        mov [si+2001],bl      ; elde varsa bir onceki adrese yazildi
        ret
topla endp

baslangic proc
        mov cx,ADET          ; her sayidaki rakam sayisi
        mov si,1000          ; ilk dizinin en soldaki rakami
        add si,cx             ; ilk dizinin en sagdaki rakami
        ret
baslangic endp

degerata proc                ; dizilere baslangic degerleri atar
dongu:
        mov ax,cx            ; her eleman (cx mod 10) ile belirleniyor
        div dl
        mov al,ah            ; kalan deger alinacak
        xor ah,ah
        mov [si],al           ; ilk diziye
        mov [si+1000],al      ; ikinci diziye
        dec si                ; bir soldaki rakam
        loop dongu
        ret
degerata endp

mesajyaz proc
        mov dx, offset mesajvar
        mov ah, 9
        int 21h
        ret
        mesajvar db "toplama islemi tamamlandi... $"
mesajyaz endp
```