

**DİKKAT:**

- Soruları, 16 bitlik 8086 mimarisine göre cevaplayınız (32 bitlik mimarideki EAX, EBX gibi yazmaçları kullanmayın.) Gerek görürseniz değişkenler tanımlayabilirsiniz. Programları yazarken yığıt bölütü tanımlanmış gibi varsayabilirsiniz.
- Kodunuz ne kadar etkin yazılmışsa, o kadar iyi puan alacaktır. Programınızı açıklayıcı yorum satırları eklemeyi unutmayın. Gereksiz yere uzatılan veya açıklama satırı olmayan kodlar, doğru çalışsa bile, eksik puan alabilir.

**Soru 1. sayi** etiketi ile verilen bir sözcüklük sayının, çift eşlik (*even parity*) yöntemine göre eşlik bitini hesaplayan ve **eslikbiti** etiketli bir byte'lık alana saklayan bir program yazınız. Örneğin, sayi=4080H değerini içeriyorsa, program sonunda eslikbiti=0 olmalıdır. (25 puan)

```
.data
    sayi dw 4080H                ;sayi= 01000000 10000000B
    eslikbiti db ?                ;Program sonunda değeri 0 olmalıdır.

.code
    mov ax, sayi
    mov cx, 16
    mov eslikbiti, 0

don:
    shl ax, 1                    ;bir bit kaydır
    jnc sifir                    ;kaydırılan bit bir mi?
    inc eslikbiti                ;bir ise eslikbiti sayısını artır
sifir:
    loop don

    and eslikbiti, 1              ;eslik bitini hesapla
    hlt
```

**Soru 2. dizgi** etiketli bir karakter dizisinde ASCII karakterlerle saklanan 4 basamaklı onluk tabandaki işaretli bir sayıyı, **sayi** etiketi ile verilen bir sözcüklük alana sayısal değer olarak aktaran bir program yazınız. Örneğin, dizgi="4138" değerini içeriyorsa, program sonunda sayi= 4138<sub>10</sub> olmalıdır. ("0" karakterinin ASCII kodu 48, "9" karakterinin ASCII kodu 57'dir.) (25 puan)

```
.data
    dizgi db "4138"          ; 4 karakter uzunluğunda bir karakter dizisi
    sayi dw ?                ; Program sonunda 413810 değerini içerecek.
    basamak dw 1000          ; Su anki basamak degeri
    taban db 10              ; 10'luk duzende basamak degeri

.code

    mov si, OFFSET dizgi
    mov basamak, 1000
    mov sayi, 0
    mov cx, 4
    mov bx, 0
don:
    mov ax, basamak

    mov bl, [si]              ;Bir basamagin ASCII degerini oku
    sub bl, 48                ;Sayi degerini hesapla
    mul bx                    ;Basamagin degerini hesapla

    add sayi, ax              ;Hesaplanan degeri sayiya ekle

    mov ax, basamak          ;Basamak degerini dusur
    div taban
    mov basamak, ax

    inc si                    ;Bir sonraki basamaga ilerle
    loop don

    hlt
```

**Soru 3. `dizgi1`** etiketli bir bellek alanında saklanan ve **`boy`** etiketli alanla uzunluęu (en fazla 20) verilen bir karakter dizisini, **`dizgi2`** adlı alana ters sırada yazdıran ve sonra **`dizgi1`** ve **`dizgi2`**'yi karşılařtırarak karakter dizisinin palindrom (tersi kendisine eşit) olup olmadığını belirleyen bir program yazınız. Karakter dizisi palindrom ise 1, deęilse 0 deęeri, **`esit`** etiketli bellek alanına aktarılmalıdır. (25 puan)

```
.data
    dizgi1 db "tabak"
    dizgi2 db 20 DUP (?)      ; Program sonunda "kabat" deęerini içermelidir.
    boy db 5
    esit db ?                ; Program sonunda 0 deęerini içermelidir.

.code

    mov si, OFFSET dizgi1
    mov di, OFFSET dizgi2
    mov dl, boy
    add di, dx
    dec di                  ;dizgi2'nin sonundan basa git.
    mov cx, dx

aktar:
    mov al, [si]            ;ters sırada dizgi1'i dizgi2'ye aktar
    mov [di], al
    inc si
    dec di
    loop aktar

    mov cx, dx
    mov si, 0

karsilastir:
    mov al, dizgi1[si]      ;ters yazilmis dizileri karsilastir
    cmp al, dizgi2[si]
    jnz esitdegil
    inc si
    loop karsilastir
    mov esit, 1             ;bütün karakterler esit
    jmp son:
esitdegil:
    mov esit, 0             ;en az bir karakter esit degil

son:
```

```
while (var1<var2 AND var2<var3)
    var2++
```

**Soru 4.** Yukarıdaki while döngüsünü 8086 çevirici dili ile gerçekleştiren kod kesimini yazınız. var1, var2 ve var3 etiketli bellek alanlarının bir byte'lık işaretli sayılar içerdiğini varsayınız. (15 puan)

don:

```
mov al,var1
cmp al,var2
jnb son
mov al,var2
cmp al,var3
jnb son
inc var2
jmp don
```

son:

**Soru 5.** Aşağıdaki kod kesimi çalıştırılırken, oluşan yazmaç değerlerini yorum satırlarında verilen boşluklara yazınız. (16 puan)

```
.data
myDouble DD 11223344h
arrayB DB 13h,37h,90h,05h
arrayW DW 5712h,0138h,9900h
.code
mov al, BYTE PTR [myDouble+2] ;AL= 22H
mov ax, WORD PTR [myDouble+1] ;AX= 2233H
mov ax, WORD PTR [arrayB+1] ;AX= 9037H
mov bl, BYTE PTR [arrayW+1] ;BL= 57H
mov ax, WORD PTR [arrayW+3] ;AX= 0001H
mov ax, 0FFFFh
mov bx, 2
imul bx ;DX= 0FFFFH AX= 0FFFEH
mov ax, 0FFFEh
idiv bl ;AH= 00H AL= 0FFH
mov ax, 0FE00h
dec al ;AX= 0FEFFH
```

**Soru 6.** Aşağıdaki kod kesimi çalıştırılırken, atlama komutlarının yanında verilen yorum satırlarına, atlamaların gerçekleşip gerçekleşmediğini, ve nedenini yazınız (6 puan)

```
mov ax,-1
cmp ax,1
ja L1 ; Atlama gerçekleşir. İşaretsiz karşılaştırmada,
... ; AX=0FFFFH değerini içerdiği için 1 değerinden daha
L1: ; büyük olarak değerlendirilir.

mov al,7Ch
test al,02h
jz L2 ; Atlama gerçekleşir. Çünkü test işleminde sonuç
... ; 0 çıktığı için Zero bayrağı kurulur ve atlama
L2: ; gerçekleşir.
```