- Aşağıdaki komutlarda kullanılan verilerin bellek adreslerini ve uzunluklarını belirtiniz.
  DS = 0100h, BX = 1500h, DI = 0300h, BASE = 0050h

KOMUT	ADRES	UZUNLUK (B)
SUB word ptr [1200h], 34h	2200h	2B
SUB DX, BASE[BX+DI]	2850h	2B
MOV AX, [DI+0500h]	1800h	2B
MOV AL, BASE[DI]	1350h	1B
ADD CX, [BX+DI]	2800h	2B
SUB AL, [BX]	2500h	1B

- Aşağıdaki komutların hatalı olup olmadıklarını belirtiniz.

KOMUT	DOĞRU	HATALI
MOV AX, [BX+1500h]	✓	
MOV CX, [DX+2000h]		✓
MOV AX, [BASE+DI]		✓
MOV DL, word ptr[BX+DI]		✓
MOV byte ptr [1500h], BASE[DI]		✓
MOV CS, [1500h]		✓
MOV AX,DS	✓	
ADD AL, CX		✓

- 0100:1000h adresinden 0100:100Fh adresine kadar bellekte saklanan 8 bitlik sayıların toplamını bulup 0100:2000h adresine kaydeden bir program yazınız. Toplama işlemleri sırasında taşma (overflow) durumu oluşursa programı sonlandırınız.

```
BASE    EQU 1000h
MOV     BX, 1000h
MOV     DI, 0000h
MOV     CX, 0010h
MOV     AL, 00h

dongu:
ADD     AL, BASE[DI]
JO      bitis
INC     DI
LOOP    dongu
MOV     BASE[BX], AL

bitis:
MOV     AH, 4Ch
INT     21h
```

- Bellekteki 0100:1000h adresinden 0100:10FFh adresine kadar depolanan bir kelimelik (word) verileri inceleyen bir program yazınız. Bu veriler içerisinde yer alan sıfırların sayısını 0100:2000h adresine, negatif sayıların sayısını 0100:2001h adresine ve pozitif sayıların sayısını 0100:2002h adresine yazınız.

```
BASE    EQU 1000h
MOV     BX, 1000h
MOV     DI, 0000h
MOV     CX, 0080h
MOV     byte ptr BASE[BX], 0000h
MOV     byte ptr BASE[BX+0001h], 0000h
MOV     byte ptr BASE[BX+0002h], 0000h

dongu:
CMP     word ptr BASE[DI], 0000h
JL      negatif
JG      pozitif

INC     byte ptr BASE[BX]
JMP     devam

negatif:
INC     byte ptr BASE[BX+0001h]
JMP     devam

pozitif:
INC     byte ptr BASE[BX+0002h]
JMP     devam

devam:
ADD     DI, 0002h
LOOP    dongu

MOV     AH, 4Ch
INT     21h
```

- 1’den 100’e kadar doğal sayıların toplamını bulup 0100:1000h adresine kaydeden bir program yazınız.

```
MOV     AX,0000h
MOV     CX,0064h

dongu:
ADD     AX,CX
LOOP    dongu

MOV     [1000h],AX

MOV     AH,4Ch
INT     21h
```

- 16x100=? işlemini hesaplayıp 0100:1000h adresine kaydeden bir program yazınız.

```
MOV     AX,0000h
MOV     CX,0064h

dongu:
ADD     AX,0010h
LOOP    dongu

MOV     [1000h],AX

MOV     AH,4Ch
INT     21h
```

- *0100:1000h* adresinden *0100:14FFh* adresi arasındaki bellek adreslerinde saklanan *16-bitlik* sayıları toplayarak elde edilen *48-bitlik* sonucu *0100:2000h* bellek adresine kaydeden programı yazınız.

```
CLD
MOV   SI, 1000h
MOV   CX, 0280h
MOV   DI, 0000h
MOV   DX, 0000h
MOV   BX, 0000h

dongu:
LODSW
ADD   BX, AX
ADC   DX, 0000h
ADC   DI, 0000h
LOOP dongu

MOV   word ptr [2000h], BX
MOV   word ptr [2002h], DX
MOV   word ptr [2004h], DI

MOV   AH,4Ch
INT    21h
```

- *N* bitlik iki sayının çarpımı *2N* bitlik bir sonuç üretmektedir. Buna göre *0100:1000h* bellek adresindeki *16-bitlik sayı ile 0100:1500h adresindeki 16-bitlik sayıyı çarparak sonucu 0100:2000h adresine kaydeden programı yazınız.*

```
MOV   DX, 0000h
MOV   AX, 0000h
MOV   BX, [1000h]
MOV   CX, [1500h]


dongu:
ADD   AX, BX
ADC   DX,0000h
LOOP  dongu
MOV   word ptr [2000h], AX
MOV   word ptr [2002h], DX

MOV   AH,4Ch
INT   21h
```

- Bellekteki *0100:1000h* adresindeki *48-bitlik* sayıdan *0100:1500h* adresindeki *8-bitlik* sayıyı çıkartarak sonucu 0100:2000h adresine kaydeden programı yazınız.

```
MOV   BX, [1000h]
MOV   DX, [1002h]
MOV   CX, [1004h]
MOV   AL, [1500h]

SUB   BL, AL
SBB   BH, 00h
SBB   DX,0000h
SBB   CX, 0000h

MOV   word ptr [2000h], BX
MOV   word ptr [2002h], DX
MOV   word ptr [2004h], CX

MOV   AH,4Ch
INT   21h
```

- Data segmentteki bir wordlük sayıların birer fazlasını extra segmente kopyalayan bir program yazınız.

```
CLD
MOV     DI,0000h
MOV     SI,0000h
MOV     CX,8000h

dongu:
LODSW
INC     AX
STOSW
LOOP    dongu
```

- Data segmentte kaç tane negatif bir wordlük sayı bulunduğunu BX'e kaydeden bir program yazınız.

```
CLD
MOV     SI,0000h
MOV     CX,8000h
MOV     BX,0000h

dongu:
LODSW
CMP     AX,0000h
JGE     pozitif
INC     BX
pozitif:
LOOP    dongu
```

- 20! İşlemini yapan program yazınız. 32-bitlik sonucu [1000h] adresinden yukarıya doğru yazınız.

```
MOV     DX,0000h
MOV     AX,0001h
MOV     BX,0002h
dongu:
        MOV     SI,DX
        MOV     DI,AX
        MOV     CX,BX
        MOV     DX,0000h
        MOV     AX,0000h
        carpma:
                ADD     AX,DI
                ADC     DX,SI
        LOOP carpma
        INC     BX
        CMP     BX,0014h
        JZ      bitis
JMP     dongu
bitis:
MOV     [1000h],AX
MOV     [1002h],DX
```

- ax+b=0 işleminde 32-bitlik a ve b sayıları sırasıyla [1000h] ve [1004h] adreslerinde bulunduğuna göre, x'i bitlik bir tamsayı olduğunu varsayarak hesaplayınız. Sonucu [2000h] adresinden yukarıya doğru yazınız.

x → DX-AX
a,b → CX-BX
ax+b → SI-DI

```
MOV     AX,0000h
MOV     DX,0000h
dongu:
    MOV   SI,0000h
    MOV   DI,0000h
    MOV   BX,[1000h]
    MOV   CX,[1002h]
    dongu_ax:
        CMP   BX,0000h
        JNZ   carpma
        CMP   CX,0000h
        JNZ   carpma
        JMP   toplam
        carpma:
        ADD   DI,AX
        ADC   SI,DX
        DEC   BX
        SBB   CX,0000h
    JMP   dongu_ax
    toplam:
    MOV   BX,[1004h]
    MOV   CX,[1006h]
    ADD   DI,BX
    ADD   SI,CX
    CMP   DI,0000h
    JNZ   siradaki
    CMP   SI,0000h
    JNZ   siradaki
    JMP   bitis
    siradaki:
    INC   DX
    ADC   SI,0000h
JMP   dongu
bitis:
    MOV   [2000h],DI
    MOV   [2002h],SI
```