

**ÖRNEK:**

**Aşağıdaki program çalıştırılıp sonlandırıldığında Akümülatör (A) içindeki son değer ne olur?**

```
MOV 20h, #10h
MOV 21h, #12h
MOV R0, #20h
MOV A, @R0
ADD A, 21h
INC A
SWAP A
END
```

Cevap: A=32h

**ÖRNEK:**

**Aşağıdaki program çalıştırılıp sonlandırıldığında Akümülatör (A) içindeki son değer ne olur?**

```
MOV A, #00h
MOV R1, #04h
Etiket-1: ADD A, R1
          DJNZ R1, Etiket-1
          INC A
          CJNE A, #11, Etiket-1
          DEC A
Son:      RL A
          END
```

Cevap: A=14h

**ÖRNEK:**

Aşağıdaki program çalıştırılıp sonlandırıldığında 78h ve 79h adreslerindeki son değerler ne olur?

```
MOV 78h, #34h
MOV 79h, #12h
MOV 7Ah, #EFh
MOV 7Bh, #12h
MOV A, 78h
ADD A, 7Ah
MOV 78h, A
MOV A, 79h
ADDC A, 7Bh
MOV 79h, A
```

Cevap: 78h=23h      79h=25h

**ÖRNEK:**

30h ile 7Fh aralığındaki verileri 1000h adresinden itibaren kopyalayan programı yazınız.

```
Etiket-1:  MOV R0, #30h
           MOV DPTR, #1000h
           MOV A, @R0
           MOVX @DPTR, A
           INC R0
           INC DPTR
           CJNE R0, #80h, Etiket-1
           END
```

**ÖRNEK:**

**1000h nolu adresten itibaren 48 adet veriyi 2000h nolu adresten itibaren kopyalayan programı yazınız.**

**1. Yol:**

```
MOV R0, #30h
MOV DPTR, #1000h
Kopyala-1: MOVX A, @DPTR
            MOV @R0, A
            INC R0
            INC DPTR
            CJNE R0, #60h, Kopyala-1
            MOV DPTR, #2000h
            MOV R0, #30h
Kopyala-2:  MOV A, @R0
            MOVX @DPTR, A
            INC R0
            INC DPTR
            CJNE R0, #60h, Kopyala-2
            RET (END)
```

**2. Yol:**

```
MOV DPTR, #1000h
MOV R0, #00h
Kopyala:    MOV DPH, #10h
            MOVX A, @DPTR
            MOV DPH, #20h
            MOVX @DPTR, A
            INC R0
            INC DPTR
            CJNE R0, #30h, Kopyala
            RET (END)
```

**ÖRNEK:**

**0450h nolu adresinden başlayarak 4Fh adet veriyi AA00h nolu adresten itibaren kopyalayan programı yazınız.**

```
MOV DPTR, #AA00h
MOV P2, #04h           ; P2 yüksek baytı R0 düşük baytı tutarak
MOV R0, #50h           ; kaydedici çifti olarak kullanabiliriz. Ve bununla
MOV R1, #4Fh           ; MOVX A, @R0 ile 0450h adresindeki veriyi alırsız.
Etiket-1: MOVX A, @R0
MOVX @DPTR, A
INC R0
INC DPTR
DJNZ R1, Etiket-1
END
```

**ÖRNEK:**

**P1.0'a bağlı bir lambayı yakıp söndüren programı yazınız.**

```
Devam: MOV P1, #00h
SETB P1.0
ACALL Bekle
CLR P1.0
ACALL Bekle
SJMP Devam
```

```
Bekle: MOV R0, #FFh           ; Lambanın belirli bir gecikmeyle yanık ya da sönmük
Dongu: MOV R1, #FFh           ; kalması için gecikme yapmamız gerekiyor
DJNZ R1, $                 ; buradaki $ işareti döngünün kendi üzerine döndüğünü
DJNZ R0, Dongu             ; gösterir.
RET
END
```

### **Çalışma Soruları:**

#### **ÖRNEK:**

30h ile 50h adresleri arasındaki verilerin toplamını bulan programı:

- a. elde oluşmadığı düşünerek, sonucunu 60h adresine,
- b. elde oluştuğunu düşünerek, sonucun yüksek baytını 60h adresine, düşük baytını 61h adresine yazacak şekilde tasarlayınız.

#### **ÖRNEK:**

30h ile 50h adresleri arasındaki verilerin en büyüğünü bularak 60h adresine yazan programı tasarlayınız.

#### **ÖRNEK:**

30h ile 50h adresleri arasındaki verilerden 3'ün katları olanları 50h adresinden itibaren kopyalayan ve yeni oluşan dizinin en büyük elemanının 3'ün kaç katı olduğunu 70h adresine kaydeden programı yazınız.

**ÖRNEK:**

Aşağıdaki program çalıştırılıp sonlandırıldığında 66h adresindeki son değerler ne olur

Dongu:

```
MOV R2, #05h
MOV A, #00h
MOV 66h, A
MOV R1, #66h
DEC 66h
SETB C
ADDC A, @R1
CPL C
ADDC A, 66h
DJNZ R2, Dongu
MOV @R1, A
END
```

Cevap: 66h= 87h

**ÖRNEK:**

	6D	6E	6F
	6A	6B	6C
+			
	66	67	68
			69

Yukarıdaki şekilde 3 baytlık iki sayının toplama işlemi görülmektedir. Hücreler içindeki sayılar ilgili baytın verisinin tutulduğu adresi göstermektedir. Birinci sayının en yüksek baytı 6Dh adresinde, birinci sayının orta baytı 6Eh adresinde, birinci sayının en düşük baytı 6Fh adresinde tutulmaktadır. İkinci sayı ve sonuç içinde durum birinci sayıya benzerdir. Bu bilgilere göre 3 baytlık iki sayının toplama işlemi gerçekleyen ve sonuçları ilgili adreslere yazan programı tasarlayınız.

BS0	DATA	6Fh	; burada adresleri karıştırmamak için adresleri ifade eden
BS1	DATA	6Eh	; değişken isim kullandık. BS= birinci sayı, IS= ikinci sayı
BS2	DATA	6Dh	; S= sonuç
IS0	DATA	6Ch	
IS1	DATA	6Bh	
IS2	DATA	6Ah	

S0 DATA 69h  
S1 DATA 68h  
S2 DATA 67h  
S3 DATA 66h

MOV S0, #00h ; sonuç değerlerini başlangıçta sıfırlanmıştır  
MOV S1, #00h  
MOV S2, #00h  
MOV S3, #00h

MOV A, BSO  
ADD A, IS0  
JNC T0  
INC S1  
T0: MOV S0, A  
MOV A, BS1  
ADD A, IS1  
JNC T1  
INC S2  
T1: ADD A, S1  
JNC T2  
INC S2  
T2: MOV S1, A  
MOV A, BS2  
ADD A, IS2  
JNC T3  
INC S3  
T3: ADD A, S2  
JNC T4  
INC S3  
T4: MOV S2, A  
RET

### ÖRNEK (Çalışma):

30h ile 4Fh adresleri arasındaki verilerden 04h'tan büyük 0Ah'tan küçük olanları 50h adresinden itibaren kopyalayan ve bu kopyalananların sayısını da 70h adresine kaydeden programı yazınız.

### P1 portunda 1 ile 9 arasındaki sayan program

```
                ORG 00H
                SJMP BASLA
                ORG 30H
BASLA:
                MOV A,#00H ;CLR A
                MOV P1,#00H
SAY:            INC A
                MOV P1,A
                CJNE A,#10,SAY
                SJMP BASLA
                END
```

### P1 portunda 1 ile 9 arasındaki sayan program fonksiyon kullanımı

```
                ORG 00H
                SJMP BASLA
                ORG 30H
BASLA: MOV R0,#00H
                MOV A,#00H ;CLR A
                MOV P1,#00H
SAY:        INC A
                MOV P1,A
                CALL TEKRAR                ; TEKRAR: DJNZ R0, TEKRAR
                CJNE A,#10,SAY
                SJMP BASLA
TEKRAR: DJNZ R0, TEKRAR ; DJNZ R0,$
                RET
                END
```

### 1 ile 5 arasındaki sayıların toplamı

```
                ORG 00H
                SJMP BASLA
                ORG 30H
BASLA:        MOV R0,#00H
                MOV A,#00H ;CLR A
SAY:          INC R0
                ADD A,R0
```



```
CJNE R0,#5,SAY
END
```

**50h ile 55h aralığındaki verileri toplayan programı yazınız**

```
ORG 00H
SJMP BASLA
ORG 30H
BASLA: MOV R0,#50H
        MOV A,#00H ;CLR A
SAY:    MOV A,@R0
        ADD A,R1
        MOV R1,A
        INC R0
        CJNE R0,#55H,SAY
END
```

**2000h ile 2005h aralığındaki verileri toplayan programı yazınız**

```
ORG 00H
SJMP BASLA
ORG 30H
BASLA:  MOV DPTR,#2000H
        MOV A,#00H ;CLR A
SAY:    MOVX A,@DPTR
        ADD A,R1
        MOV R1,A
        INC DPTR
        MOV R7,DPL
        CJNE R7,#05H,SAY
END
```

**30h ile 7Fh aralığındaki verileri 1000h adresinden itibaren kopyalayan programı yazınız.**

```
MOV R0, #30h
MOV DPTR, #1000h
Etiket-1: MOV A, @R0
          MOVX @DPTR, A
          INC R0
          INC DPTR
```

```
CJNE R0, #80h, Etiket-1
END
```

**1000h nolu adresten itibaren 48 adet veriyi 2000h nolu adresten itibaren kopyalayan programı yazınız.**

**1. Yol:**

```
MOV R0, #30h
MOV DPTR, #1000h
Kopyala-1: MOVX A, @DPTR
            MOV @R0, A
            INC R0
            INC DPTR
            CJNE R0, #60h, Kopyala-1
            MOV DPTR, #2000h
            MOV R0, #30h
Kopyala-2:  MOV A, @R0
            MOVX @DPTR, A
            INC R0
            INC DPTR
            CJNE R0, #60h, Kopyala-2
            RET (END)
```

**2. Yol:**

```
MOV DPTR, #1000h
MOV R0, #00h
Kopyala:   MOV DPH, #10h
            MOVX A, @DPTR
            MOV DPH, #20h
            MOVX @DPTR, A
            INC R0
            INC DPTR
            CJNE R0, #30h, Kopyala
            RET (END)
```

**0450h nolu adresinden başlayarak 4Fh adet veriyi AA00h nolu adresten itibaren kopyalayan programı yazınız.**

```
MOV DPTR, @AA00h
MOV P2, #04h      ; P2 yüksek baytı R0 düşük baytı tutarak
MOV R0, #50h      ; kaydedici çifti olarak kullanabiliriz. Ve bununla
MOV R1, #4Fh      ; MOVX A, @R0 ile 0450h adresindeki veriyi alırız.
```

Etiket-1:      MOVX A, @R0  
                 MOVX @DPTR, A  
                 INC R0  
                 INC DPTR  
                 DJNZ R1, Etiket-1  
                 END

**P1.0'a bağlı bir lambayı yakıp söndüren programı yazınız.**

Devam:        MOV P1, #00h  
                 SETB P1.0  
                 ACALL Bekle  
                 CLR P1.0  
                 ACALL Bekle  
                 SJMP Devam

Bekle:        MOV R0, #FFh            ; Lambanın belirli bir gecikmeyle yanık ya da sönük  
Dongu:        MOV R1, #FFh            ; kalması için gecikme yapmamız gerekiyor  
                 DJNZ R1, \$            ; buradaki \$ işareti döngünün kendi üzerine döndüğünü  
                 DJNZ R0, Dongu        ; gösterir.  
                 RET  
                 END

