

MIKRODENETLEYICILER

Ege Üniversitesi Ege MYO
Mekatronik Programı

BÖLÜM 3

Assembler Programlama ve Program Geliştirme

Program Geliştirme

- ◆ Problem Tanımlama
- ◆ Bağlantı Şekli
- ◆ Algoritma
- ◆ Akış Diyagramı
- ◆ Kaynak Program Yazma
- ◆ Assembly
- ◆ Simulatör
- ◆ Mikroişlemciye Yükleme ve Çalıştırma

Problem Tanımlama

- ◆ Problem kullanılacak denetleyiciye veya işlemciye göre teknik bir dille yeniden tanımlanır.
- ◆ Derste verilen sorular genellikle tanımlanmış problemler olduğu için yeniden tanımlanmaz.

Bağlantı Şekli

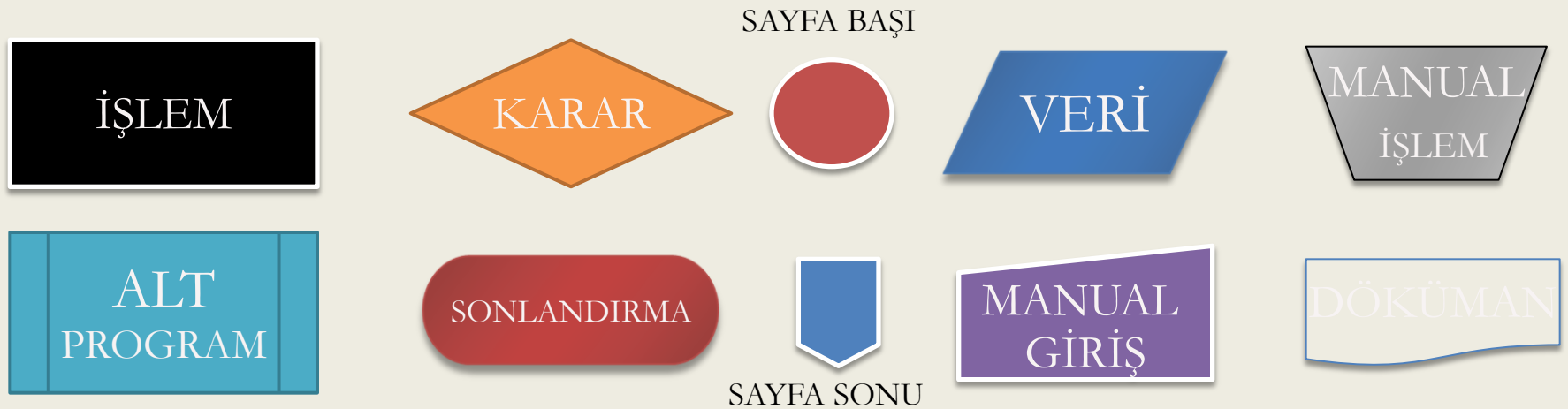
- ◆ Eğer tanımlanan problem yeni bir bağlantı gerekiyorsa açık veya blok şema çizilir.
- ◆ Özellikle bağlantı yapılacak port isimleri mutlaka yazılır.
- ◆ Bağlanacak elemanların hatlarının isimleri yazılmalıdır.

Algoritma

- ◆ Çözüm yönteminin konuşma dili ile maddeler halinde tanımlanmasıdır.
- ◆ Algoritma problemin çözümünü temel oluşturur.
- ◆ Bağlantı şeklini desteklemelidir.
- ◆ Yalın bir dil kullanılarak yazılmalıdır.
- ◆ Bir problemin birden fazla çözüm algoritması olabilir.

Akış Diyagramı

- ◆ Algoritmanın şekilsel olarak gösterimidir.
- ◆ Her yapılan işlem bir simge ile gösterilir.
- ◆ Programın işleyiş sırasını anlamayı kolaylaştırır.



Kaynak Program

;Programın başlığı, yazan kişi, tarih

;program hakkında açıklama

Org 0

isim:

Mov A,#25

; program buradan başlar

;A=25

.....

;
.....

TKR:

.....

;
.....

Sjmp \$

;son

end

;program burada biter.

Simulator

- ◆ Yazılan program assebmly edilir.
- ◆ Simulatöre yüklenerek programın çalışması adım modunda denetlenir.
- ◆ Hata var ise algoritmaya geri dönülür.
- ◆ Hatalar tamamen yok edilinceye kadar devam edilir

Mikroişlemciye Yükleme ve Çalıştırma

- ◆ Hataları giderilen program programlayıcı kullanılarak mikroişlemciye yüklenir.
- ◆ Gerekli bağlantılar kart üzerinde ve çevresinde yapılır.
- ◆ Devreye güç verilerek çalıştırılır ve sonuçlar gözlemlenerek doğruluğu belirlenir.
- ◆ Hata var ise algoritmaya geri dönülür.
- ◆ Hata yok ise tasarlanan cihaz üretime hazırdır.

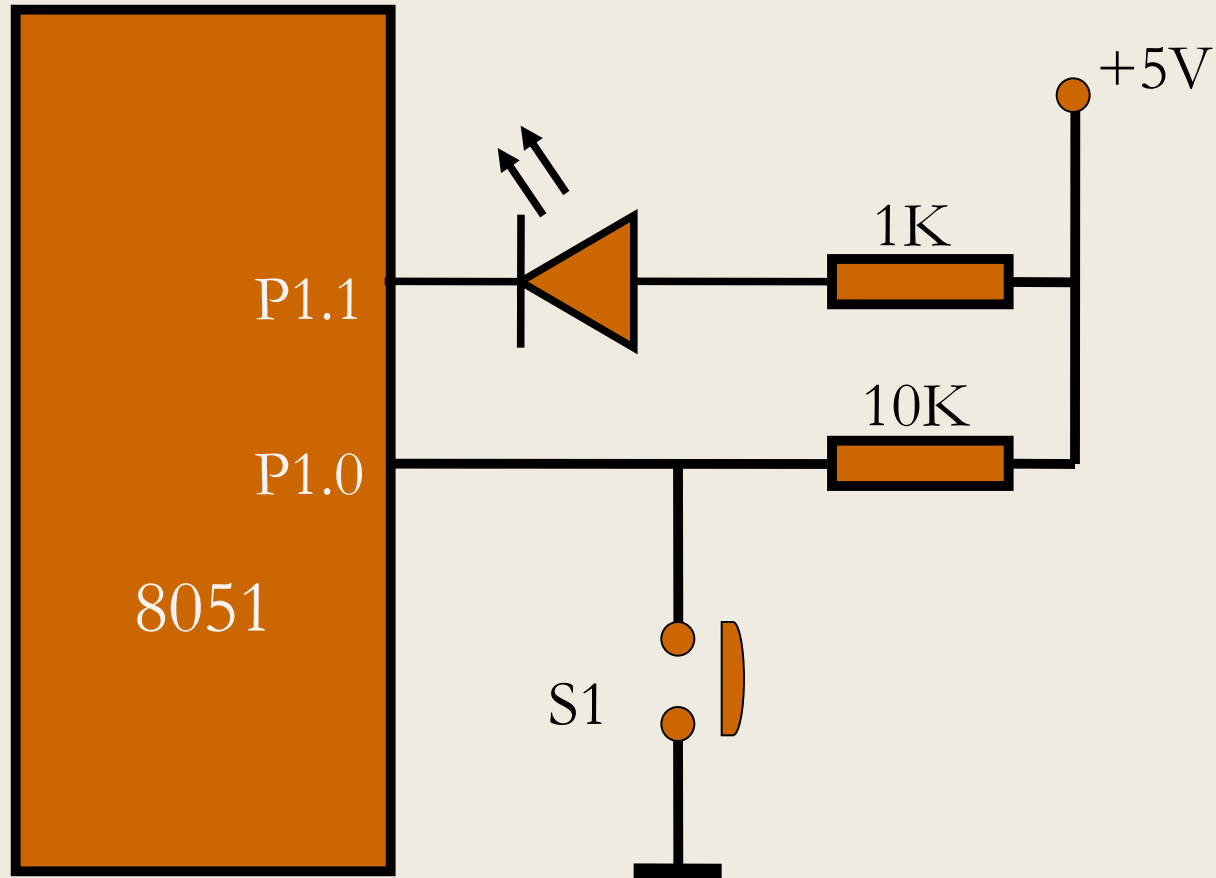
Örnek

- ◆ Bir butona basıldığında LED'i yakan ikinci basmada ise söndüren programı yazın, sistemi kurup çalıştırın.

1. Problemi Yeniden Tanımlama:

8051'in port hattına bir buton ve bir diğer hattına LED bağlıdır. Butona her basıldığında LED'in durumunu tersleyen programı yazın.

2. Bağlantı Şekli

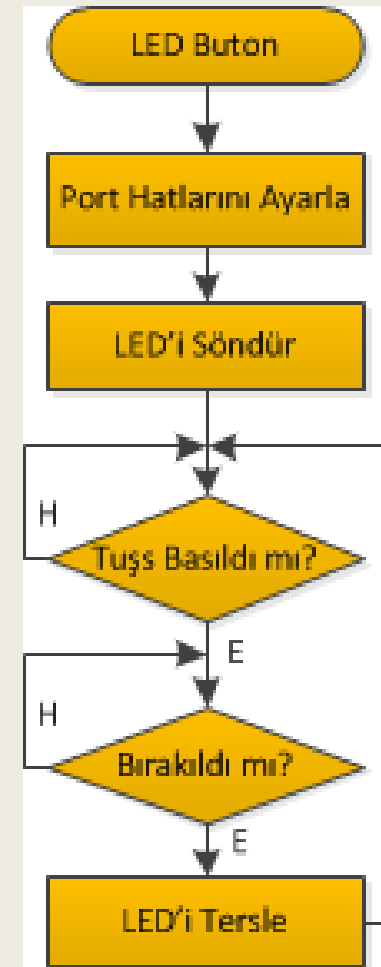


Program Geliştirme

3. Algoritma

1. Buton için port hattını giriş yap.
2. LED için port hattını çıkış yap
3. LED'i başlangıç için söndür.
4. Butona basılana kadar bekle.
5. Buton bırakılana kadar bekle.
6. LED'i tersle.
7. Adım 4'e dön.

4. Akış Diyagramı



5. Kaynak Program

Org 0

BUT_LED:

Mov P1,#0FFh ;P1.0 giriş, P1.1 çıkış, LED sönük

BUT:

Jb P1.0,\$;Butona basılana kadar bekle

Jnb P1.0,\$;Buton bırakılana kadar bekle

Cpl P1.1 ;LED'i tersle

Sjmp BUT ;Sürekli yap

end

Örnek 1 16 Bit Değişkene 8 Bit Değişken Toplama.

Bu alt program A'yı DPTR'ye toplar, taşma var ise elde bayrağı kurulur. İşletilmeden önce DPTR'ye ve A'ya bir değer girilmelidir.

İşlem sonunda; $DPTR = DPTR + A$

Etkilenen yazaçlar; PSW.CY, DPTR

topla16b:

<i>Push Acc</i>	<i>; A'yı yedekle.</i>
<i>Add A,DPL</i>	<i>; A'ya DPL topla.</i>
<i>Mov DPL,A</i>	<i>; Sonucu DPL'ye yaz.</i>
<i>Mov A,DPH</i>	<i>; DPH'yi al.</i>
<i>Addc A,#00h</i>	<i>; Elde var ise YDB'ı artır.</i>
<i>Mov DPH,A</i>	<i>; DPH yaz.</i>
<i>Pop Acc</i>	<i>; A'yı yedekten al.</i>
<i>Ret</i>	<i>; Ana programa geri dön.</i>

ÖRNEK 2 16 Bit Değişkenden 8 Bit Değişken Çıkarma.

;Bu alt program DPTR yazacından Acc'yi çıkarır. Sonuç negatif ise elde
;bayrağını kurar. İşletilmeden önce DTPR'ye ve Acc'ye bir değer girilmelidir.
;İşlem sonunda DPTR=DPTR-Acc
;Etkilenen yazaçlar; PSW.CY, DPTR

cikar16b:

<i>Push Acc</i>	<i>; A'yı yedekle.</i>
<i>Clr c</i>	<i>; Eldeyi temizle</i>
<i>Xch a,DPL</i>	<i>; A ile DPL'yi yer değiştir.</i>
<i>Subb a,DPL</i>	<i>; Çıkar.</i>
<i>Mov DPL,a</i>	<i>; DPL'yi yaz</i>
<i>Mov a,DPH</i>	<i>; DPH'yi al.</i>
<i>Subb a,#00h</i>	<i>; Borç var ise YD kısmı bir azalt.</i>
<i>Mov DPH,a</i>	<i>; DPH'yi yaz.</i>
<i>Pop Acc</i>	<i>; A'yı yedekten al</i>
<i>Ret</i>	<i>; Ana programa geri dön</i>

ÖRNEK 3 16 Bitlik Çıkarma

;Bu program 16 bit sayıdan 16 bit sayıyı çıkarır.

;R7R6-R5R4=R3R2

Cıkar16bit:

MOV A,R6 ;1. değişkenin düşük kısmını A'ya al

CLR C ;Eldeyi temizle

SUBB A,R4 ;2. değişkenin düşük kısmını A'dan çıkar

MOV R2,A ;Sonucun düşük kısmını yaz

MOV A,R7 ;1. değişkenin yüksek kısmını A'ya al

SUBB A,R5 ;2. değişkenin yüksek kısmını A'dan çıkar

MOV R3,A ;Sonucun yüksek kısmını yaz

RET ;Sonucu R2 ve R3'te ana programa götür.

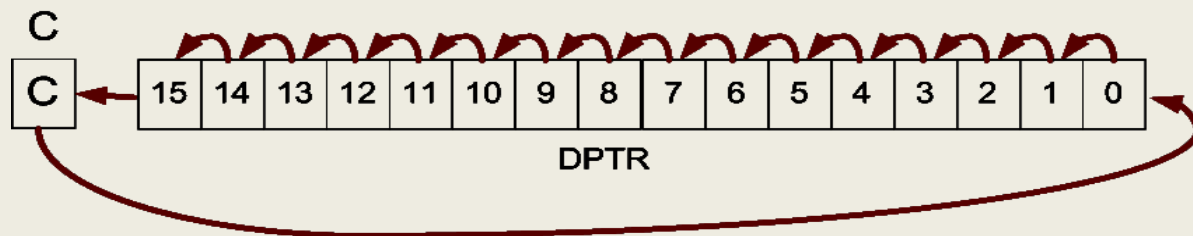
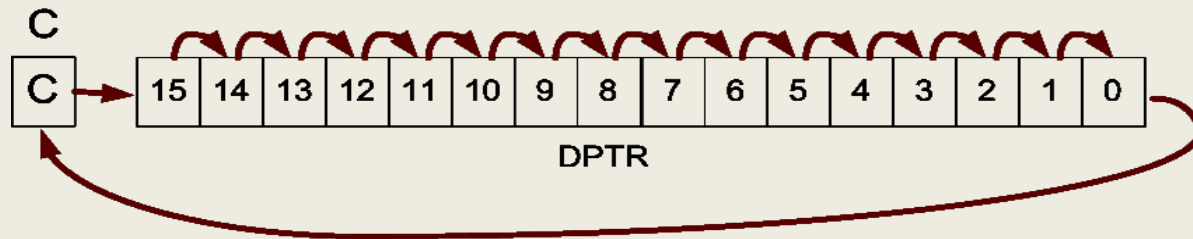
ÖRNEK 4 DPTR Yazacını Bir Bit Sağa Öteleme.

/ Bu alt program DPTR yazacının içeriğini aritmetik bir bit sağa öteler.
İşletilmeden önce DPTR içerisine bir değer girilmeli. */*

Dptrsag:

<i>Push Acc</i>	<i>; A'yı yedekle.</i>
<i>Clr C</i>	<i>; Eldeyi temizle.</i>
<i>Mov A,DPH</i>	<i>; DPTR'nin yüksek kısmını al.</i>
<i>Rrc A</i>	<i>; Sağa ötele.</i>
<i>Mov DPH,A</i>	<i>; DPH'ye yaz.</i>
<i>Mov A,DPL</i>	<i>; DPTR'in düşük kısmını al.</i>
<i>Rrc A</i>	<i>; Sağa ötele.</i>
<i>Mov DPL,A</i>	<i>; DPL'yi yaz.</i>
<i>Pop Acc</i>	<i>; Acc'yi tedekten al.</i>
<i>Ret</i>	<i>; Ana programa dön.</i>

RLC A, RRC A



Örnek 5: DPTR Yazacını Bir Bit Sola Öteleme.

;Bu alt program DPTR yazacının içeriğini bir bit sola öteler.

;DPTR * 2 = DPTR

Dptrsol:

<i>Push Acc</i>	<i>; Acc'yi yedekle.</i>
<i>Clr C</i>	<i>; Eldeyi temizle.</i>
<i>Mov A,DPL</i>	<i>; DPTR'nin düşük kısmını al.</i>
<i>Rlc A</i>	<i>; Sola ötele.</i>
<i>Mov DPL,A</i>	<i>; DPL'ye yaz.</i>
<i>Mov A,DPH</i>	<i>; DPTR'nin yüksek kısmını al.</i>
<i>Rlc A</i>	<i>; Sola ötele.</i>
<i>Mov DPH,A</i>	<i>; DPH'ye yaz.</i>
<i>Pop Acc</i>	<i>;Yedekten al</i>
<i>Ret</i>	<i>; Ana programa dön.</i>

Örnek 6 DPTR'nin 10 Katını Alma.

*;Bu alt program DPTR'nin 10 katını alır. DPTR'e işlem öncesi bir 0-6553 arası bir değer yazılmalıdır. İşlem sonunda DPTR=DPTR * 10, etkilenen yazaçlar PSW,DPTR*

DPTRX10:

Push B	<i>; B'yi yedekle.</i>
Push Acc	<i>; A'yı yedekle</i>
Mov A,DPL	<i>; Düşük kısmı al</i>
Mov B,#10	<i>; Çarpanı B'ye yaz</i>
Mul AB	<i>; BA = DPL * 10</i>
Mov DPL,A	<i>;Sonucun düşük kısmını DPL yaz</i>
Push B	<i>;Sonucun yüksek kısmını yedekle</i>
Mov A,DPH	<i>;Yüksek kısmını al</i>
Mov B,#10	<i>;Çarpanı B'ye yaz</i>
Mul AB	<i>;BA = DPH * 10</i>
Pop B	<i>;B yedekten al</i>
Add A,B	<i>;Düşük kısımdan oluşan ile topla</i>
Mov DPH,A	<i>;Sonucun yüksek kısmını DPH yaz</i>
Pop Acc	<i>;A'yı yedekten al</i>
Pop b	<i>;B'yi yedekten al</i>
ret	<i>;Ana programa dön</i>

Örnek 7 DPTR'yi 4 Bit Sağa Ötele

;Bu Alt Program Dptr'yi 4 Bit Sağa öteler. Etkilenen Yazaçlar Psw,Dptr, DPTR'yi 16'ya böler.

Dptr4sag:

	<i>Push Acc</i>	<i>; A'yı Yedekle</i>
	<i>Push 0</i>	<i>; R0'ı Yedekle</i>
	<i>Mov R0,#4</i>	<i>; Öteleme Sayacını Ayarla</i>
<i>L1:</i>	<i>Mov A,DPH</i>	<i>; Yüksek Kısmı Al</i>
	<i>Clr C</i>	
	<i>Rrc A</i>	<i>; Sağa ötele</i>
	<i>Mov DPH,A</i>	<i>; DPH'ye Yaz</i>
	<i>Mov A,DPL</i>	<i>; Dptr'nin düşük Kısmını Al</i>
	<i>Rrc A</i>	<i>; Sağa ötele</i>
	<i>Mov DPL,A</i>	
	<i>Djnz R0,L1</i>	<i>; Sayaç=0 olana kadar ötele</i>
	<i>Pop 0</i>	<i>; Yedekten Al</i>
	<i>Pop Acc</i>	<i>; Yedekten Al</i>
	<i>Ret</i>	<i>; Ana Programa Geri Dön</i>

Örnek 8 Dptr 4 Bit Sola Öteleme

;Bu Alt Program Dptr Yazacını 4 Bit Sola Öteler, Düşük Değerli Bitlere Sıfır Yerleştirir.

;Etkilenen Yazıcılar; Psw.Cy, Dptr

Dptr4sol:

```
L2:      Push Acc      ; A'yı Yedekle  
         Push 0       ; R0'ı Yedekle  
         Mov R0,#4    ; Öteleme Sayacını Kur  
         Clr C        ; Öteleme İçin Sıfırla  
         Mov A,Dpl    ; Dptr'nin Düşük Kısmını Al  
         Rlc A        ; Sola Ötele  
         Mov Dpl,A    ; Dpl'yi Yaz  
         Mov A,Dph    ; Dptr'nin Yüksek Kısmını Al  
         Rlc A        ; Sola Ötele  
         Mov Dph,A    ; Dph'yi Yaz  
         Djnz R0,L2   ; Sayacı Sıfır Olana Kadar Devam Et  
         Pop 0       ; Yedekten Al  
         Pop Acc     ; Yedekten Al  
         Ret         ; Ana Programa Geri Dön
```

Örnek 9 AND Maskeleye ve Tablo Okuma

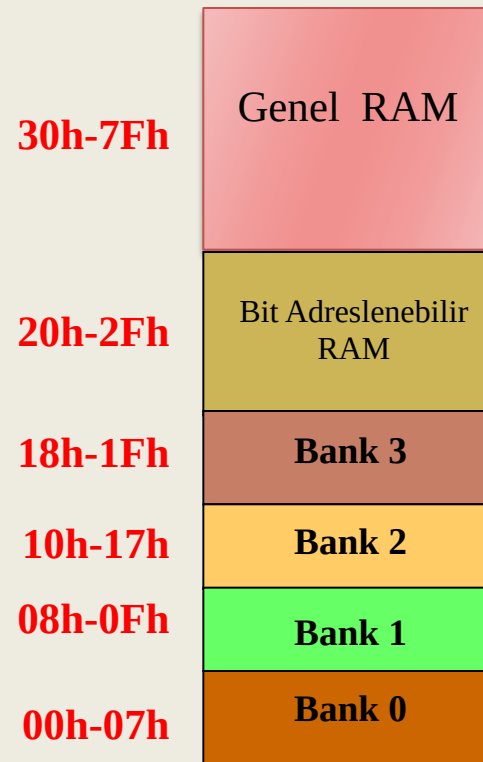
P1'den okunan 0 ile F arası sayıların karesini başvuru tablosu kullanarak alan ve P2'ye yazan programı yazın

Kare_al:

```
Mov DPTR, #TABLO ; Tablonun başlangıç adresi
Mov P1, #0FFH ; P1'i giriş yap
TKR: Mov A, P1 ; Sayıyı P1'den oku
Anl A, #0Fh ; Yüksek değerli 4 biti sıfırla
Movc A, @A+DPTR ; Sayının karesini tablodan al
Mov P2, A ; Sonucu P2'ye yaz
Sjmp TKR ; Sürekli yap
ORG 300H ; Tablo 0300H'dan başlar
TABLO: DB 0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169,
196, 225
```


8051'de Yığın Kullanımı

- ◆ Yığının **SP** (stack pointer) yazacı ile erişebiliriz.
- ◆ Yığın işaretleyici SP 8051'de 8 bit genişliğindedir, bunun anlamı bu yazaç içeriği 00'dan FFh kadar değerleri içerebilir.
- ◆ 8051 çalıştırıldığında, SP yazacının içeriği 07'dir.



Yığın reset sonrası buradan başlar

Yığın????



Altprogramların Kullanımı

ANA_PROG:

Org 0

Mov SP,#7Fh	;yığını üst 128 bayta ötele
Mov A,#25H	;A=25H
Acall topla16b	;toplama altprogramını çağır
Acall Cikar16bit	;Çıkarma altprogramını çağır
Mov DPTR,#2551	;DPTR=2551
Acall DPTRX10	;DPTR*10
Acall Dptr4sol	;DPTR*2*2*2*2
Acall Cikar16bit	;Çıkarma altprogramını çağır
Sjmp \$	

Ödev - 4

- ▶ Bölüm 4'ün sorularını yanıtlayınız.
- ▶ (Haftaya ders saatinde teslim edilecek)