

Çalışma soruları 2

1: p1.0 pininden 0 değeri gelmiş ise P3'e bağlı 8 ledin durmaksızın (bir döngü içerisinde) sadece sağdan sola doğru (p3.0 dan p3.7'ye doğru) teker teker sırasıyla yanmasını gerçekleştiren kodu yazınız. (rl veya rlc komutunu bir döngü içinde kullanmadan yapınız.)

Açıklama : jb veya jnb ile p1.0 testi yap. P1.0 = 0 ise 7 kez rl kullan.

2: ilk soruda rl komutunu 1 kez kullanarak döngü içinde kodu tekrar yazınız.

Açıklama : jb veya jnb ile p1.0 testi yap. P1.0 = 0 ise bir döngü içinde 1 kez rl kullan. Ancak döngü 7 lik olacak.

3: Bu sefer p1.0 pininden 0 değeri gelmiş ise soldan sağa, p1.1'den de 0 değeri gelmiş ise sağdan sola ledleri tek tek sırasıyla yakınız.

4: ilk önce e ve f yi yap.

a) CJNE komutu kullanarak 5! (faktöriyel $5=5*4*3*2*1$) işlemini gerçekleştiriniz. Sonucu P0'da gözlemleyiniz.

b) CJNE komutu kullanarak $(5+4+3+2+1)$ işlemini gerçekleştiriniz. Sonucu P0'da gözlemleyiniz.

c) CJNE komutu kullanarak 5! (faktöriyel $5=1*2*3*4*5$ şeklinde) işlemini gerçekleştiriniz. Sonucu P0'da gözlemleyiniz.

d) CJNE komutu kullanarak $(1+2+3+4+5)$ işlemini gerçekleştiriniz. Sonucu P0'da gözlemleyiniz.

e) DJNZ komutu kullanarak 5! (faktöriyel $5=5*4*3*2*1$) işlemini gerçekleştiriniz. Sonucu P0'da gözlemleyiniz.

f) DJNZ komutu kullanarak $(5+4+3+2+1)$ işlemini gerçekleştiriniz. Sonucu P0'da gözlemleyiniz.

5: CJNE komutu ile aşağıda verilen adreslere dolaylı adresleme kullanarak karşılarındaki verileri yazınız. (Tabloda 16 adres ve veri vardır)

Adres	Veri
1FH	4FH
1EH	4EH
.	.
.	.
11H	41H
10H	40H

6: CJNE komutu ile aşağıda verilen adreslere dolaylı adresleme kullanarak karşılarındaki verileri yazınız. (Tabloda 16 adres ve veri vardır)

Adres	Veri
1FH	70D
1EH	68D
1DH	66D
.	.
.	.
11H	42D
10H	40D

7: P1 bağı 4 button vardır (P1.0-P1.1-P1.2-P1.3). (Butonlara basınca, pine gelen değerler 0 olmaktadır.) Dolayısıyla hiçbir pine basılmamışsa P1'den gelen değer 15 olmaktadır. Yüksek Nibble'dan gelen veriler maskeleyle sıfırlanmalıdır. P1'den bu şekilde gelecek verinin 1'den gelen değere kadar toplamını p3'te gözlemleyiniz. (Örneğin porttan gelen veri 10 ise 1'den 10'a kadar olan sayıların toplamını p3'te gözlemleyiniz.)

8: p3'ten gelen veri p0'dan gelen veriye eşit ise p1.1 bağlı ledi, aksi takdirde p1.2 ledini yakınız. JZ komutunu kullanınız.

9:

P2-P3= işleminin sonucu pozitif ise p1.1=5v yapınız aksi takdirde p1.2=5v yapınız.

(P2 ve P3 portlardır.)

10: Aşağıda verilen 16 bitlik sayıları çıkarınız. Sonuç pozitif ise p1.1=5v yapınız aksi takdirde p1.2=5v yapınız.

1105h- 080Ah

11.

Tablo1

Adres	Veri (decimal)
10h	10
11h	9
12h	8
13h	7
14h	6
15h	5
16h	4
17h	3
18h	2
19h	1

- **Tablo1'de** verilen şekilde **RAM'e belirtilen adreslere karşılık gelen verileri bir döngü ile** kaydediniz.
- Ardından adreslerdeki verilerin **bilinmediği** varsayılarak verilerin toplamını döngü ile gerçekleyiniz (döngü için **cjne** komutunu kullanınız), **P2'ye toplam sonucunu yazınız** ve sonsuz döngüde bekleyiniz.

12.

Tablo2

İkilik çarpım tablosu	Kaydedilecek RAM adresleri
2x1=2	00h
2x2=4	01h
.	.
.	.
.	.
.	.
2x9=18	08h

- Yanda verilen **Tablo 2'de** çarpım tablosunda 2 ile çarpma işlemlerinin karşılıklarını RAM'in 00h-08h adresleri arasında **yukarıya doğru artacak şekilde** döngü kurarak kaydediniz ve işlem bitince sonsuz döngüde bekleyiniz.
- Döngü için **djnz** komutunu kullanınız.
- 2 ile çarpma işlemleri mikrodnetleyiciye yaptırılacak, doğrudan sonuçlar yazdırılmayacaktır.

13.

- Aşağıda Tablo 3'te verildiği biçimde, çarpım tablosunun üç, dört ve beş ile çarpma işlemlerinin sonuçlarını Tablo 3'te belirtilen RAM bölgelerine kaydediniz.
- İşlemlerin **içi içe döngü** mantığında gerçekleştirilmesi zorunludur.
- Tüm çarpma işlemleri mikrodenetleyiciye yaptırılacak, doğrudan sonuçlar yazdırılmayacaktır.
- (Döngüler için cjne komutunu kullanınız)

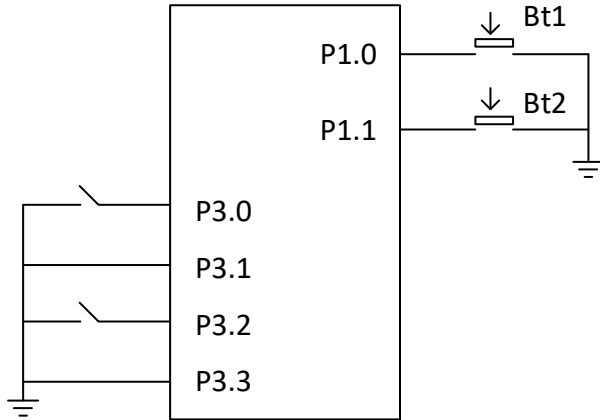
Tablo 3

İkilik çarpım tablosu	Kaydedilecek RAM adresleri	Üçlük çarpım tablosu	Kaydedilecek RAM adresleri	Dörtlük çarpım tablosu	Kaydedilecek RAM adresleri
3x1=3	10h	4x1=4	1Ah	5x1=5	24h
3x2=6	11h	4x2=8	1Bh	5x2=10	25h
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
3x10=30	19h	4x10=40	23h	5x10=50	2Dh

Çalışma soruları 3

1. P3'te yüksek nibble'daki değer bilinmiyor. P3'ün düşük nibble'da 10d değeri vardır. Bt1 butonuna basıldığında p3'ten alınan düşük nibble değer (10d=0Ah) 2 katını, Bt2 butonuna basıldığında ise 3 katını alınız. Sonucu P0'da gözlemleyiniz.

Pinlerin kontrolünü (butonlara basılıp basılmadığını) jb veya jnb komutu kullanarak yapılacaktır.



2. Yukarıdaki çarpma işlemlerini alt programda yapınız. Bir alt programınız olacak o sadece 2 katını alacak, diğer bir alt programınız ise 3 katını alacak.

3. 2. Sorudaki programı, Keil programını adım adım çalıştırırken acall kodu çalışmadan önce döneceğiniz adresi belirleyiniz. Sp değişimleri ile sp adreslerin gösterdiği içeriklerini gözlemleyiniz.

4. Aşağıdaki programı yazınız. Veritabanı etiketinin program adresini gözlemleyiniz. Ayrıca veri tabanındaki değerleri ve bunların program hafızadaki adreslerini gözlemleyiniz. Program adresi hem debug modunda dissassembly penceresinde hemde memory kısmında c:0x00 şeklinde gözlemleyebilirsiniz.

```
cs1.a51
1  #include "aduc841.h"
2  org 0000
3  sjmp ayar
4  ayar:
5      ; burada mikro
6      ; ayarlari yapilir
7  basla:
8      mov r0,#04h
9      mov a,#33h
10
11     sjmp basla
12
13 veritabani: db 19d,
14              8d,
15              17d,
16              6d,
17              15d,
18              4d,
19              13d,
20              2d,
21              11d,
22              0d // 10 adet veri
23 ;veritabani: db 19d,8d,17d,6d,15d,4d,13d,2d,11d,0d // bu sekilde daha rahat olusturulabilirdi.
24 end
```

Bundan sonraki sorularda veri tabanındaki değerler bilinmediği sadece veri tabanındaki veri sayısının 10 olduğu bilinerek aşağıdaki soruları cevaplayınız.

5. 4. Soruda hazırladığınız programı şu şekilde güncelleyelim: Veri tabanındaki değerlerin bilinmediği sadece veri tabanında 10 veri olduğu bilgisi ile veri tabanındaki 8. değeri (yani 1d verisini) p0 da gösteriniz.

6. 4. Soruda hazırladığınız programı şu şekilde güncelleyelim: Bir 10 luk döngü ile (**djnz**) p0 da tek tek veri tabanındaki bilgileri aşağıda verilen a ve b adımlarında verilen şekilde p0 da gözlemleyiniz.

a. İlk önce veri tabanındaki 10. veriden başlayarak geriye doğru 0. veriye kadar olan verileri p0 da tek tek gözlemler. (0d->p0'da, ardından 11d->p0'da, ardından 2d->p0'da.....son olarak 19d->p0'da olacak şekilde)

b. Yine djnz kullanarak 0. veriden 10. veriye kadar olan verileri tek tek p0'da gözlemler.

7. 6. Sorudaki kodu cjne ile hem a hem de b şıkkı için yapınız. Bu sefer b şıkkı kolaylıkla yapılabilir.

8. 4. Soruda hazırladığınız programı şu şekilde güncelleyelim: Veri tabanında bir arama yapma programı yapılacaktır. Örneğin veri tabanında 6d değeri var ise p0.1 yakınız, yok ise p0.2 yakınız. Örneğin veri tabanında 16d değeri var ise p0.1 yakınız yok ise p0.2 yakınız.

9. 4. Soruda hazırladığınız programı şu şekilde güncelleyelim: Bir döngü ile veri tabanındaki tek indisli bilgileri tek tek p0'da gözlemleyiniz. Yani 1. 3. 5. 7. ve 9. sıradaki bilgileri

10. 4. Soruda hazırladığınız programı şu şekilde güncelleyelim: Bir döngü ile veri tabanındaki tek indisli bilgileri toplayalım ve sonucu p0'da gözlemleyelim.

11. 4. Soruda hazırladığınız programı şu şekilde güncelleyelim: Bir döngü ile veri tabanındaki tek indisli bilgileri toplayalım ve sonucu r7'ye yazalım, ardından çift indisli bilgileri toplayalım ve sonucu r6'ya yazalım. Ve r7-r6 çıkartma işlemini gerçekleştiriniz. Sonuç pozitif ise p0.1 deki ledi yakınız, negatif ise p0.2deki ledi yakınız.