

<b>SINIF NO:</b>	<b>EEM 304</b> <b>MİKROİŞLEMCİLER</b> 2017 – 2018 Bahar Yarıyılı  <b>VİZE</b> <b>06.04.2018</b> Süre: 90dk
<b>Ad:</b>	
<b>Soyad:</b>	
<b>Öğrenci No:</b>	
<b>İmza:</b>	

1) Programın bir kez döngüsünü tamamlaması (sjmp devam) aşamasına kadarlık kısmı göz önüne alınarak, karşılaşılan her alt programa ait komutta (**ACALL, LCALL, RET**) programın **nereye dallediğini kod metni üzerinde okla gösteriniz** ve bunu sırayla (1, 2, 3,...) numaralandırınız. Ayrıca RAM'in değişen tüm hücrelerinin; SP, A ve DPTR kaydedicilerine ait tüm değişimleri aşağıdaki şablonlar üzerinde belirtiniz. Bu işlemler dallanma numaralarına bağımlı olarak yapılmayacaktır. RAM hücrelerindeki değişimleri soldan sağa, diğer saklayıcılarda yukarıdan aşağıya doğru gösteriniz.

Adres:		
main:		
0000h	mov	DPTR,#001Fh
0003h	mov	SP,#70H
devam:		
	mov	acc,#40h
	mov	07h,#0FFh
	push	DPL
	push	DPH
0010h	acall	hesap
	mov	DPTR,#22FFh
1 0015h	mov	acc,#0FFh
	nop	
0019h	sjmp	devam
hesap:		
001Bh	setb	rs0
	mov	r0,#12h
	mov	r1,#13h
	mov	0Ah,#14h
	mov	0Bh,#15h
	push	0Bh
	mov	acc,#00h
	push	acc
	nop	
003Fh	ret	
0040h	end	

RAM				
Değerler (Her kutuya bir duruma karşılık değer)				Adres
00				76
15				75
00				74
12				73
00				72
17				71h
				70h
15				0B
14				0A
13				09
12				08
FF				07h

SP	A
07	40
70	00
71	FF
72	
73	
74	
75	
76	
75	
74	

DP
001F

Başarılar dileriz...

2) Aşağıdaki program kodunun çalışması durumunda A, PSW, SP kaydedicilerinde ve RAM’de olan tüm değişiklikleri ve ilgili hücrelerin son değerlerini gösteriniz.

Açıklama:

- RAM’i derste anlatıldığı biçimde en altı 00h adresi olacak biçimde planlayınız. Adresleri kutu dışına sağa yazınız. Değişen değerleri soldan sağa yazınız.
- Diğer 3 kaydedicide her bir satırı bir duruma/komuta karşılık gelecek biçimde kullanınız. Bu kaydediciler için ilk değeri en üste yazarak yukarıdan aşağıya sıra takip ediniz.
- PSW’da yalnızca RS0, RS1 ve C’deki değişimler dikkate alınacaktır.
- Tüm hücrelerde değişen değerlerin üzerini rahatlıkla okunabilecek şekilde çiziniz. Üzeri çizilmeyen değer son değer olacaktır.

		RAM			
clr	a				
mov	r0, #20h				
mov	@r0, #022h				
mov	09h, #21h				
setb	rs0				
mov	@r1, #044h				
setb	09h				
mov	12h, #13h	44H / 46H	21H		
mov	13h, #12h	22H / A2H	20H		
mov	a, 13h				
subb	a, 12h				
anl	psw, #0E7h				
mov	07h, c				
clr	c				
dec	09h				
setb	psw.3	12H	13H		
mov	a, r1	13H	12H		
rr	a				
rrc	a				
mov	dp1r, #kareal				
movc	a, @a+dp1r				
nop					
		21H/20H	09H		
		20H	00h		

C			RS1	RS0			
PSW							
0	X	X	0	0	X	X	X
0	X	X	0	1	X	X	X
1	X	X	0	1	X	X	X
1	X	X	0	0	X	X	X
0	X	X	0	0	X	X	X
0	X	X	0	1	X	X	X
	X	X			X	X	X

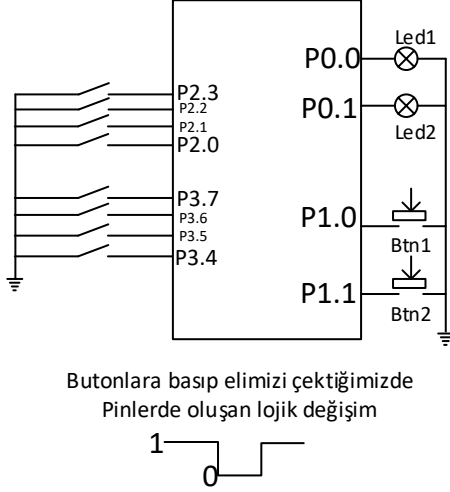
A	
00H	
12H	
FFH	
20H	
10H	
08H	
40H=64d	

kareal:db 0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100

END

SINIF NO:	EEM 304	
Ad:	MİKROİŞLEMCİLER	
Soyad:	2017 – 2018 Bahar Yarıyılı	
Öğrenci No:	VİZE	
İmza:	06.04.2018	
	Süre: 90dk	

3)



Aşağıdaki iki şık birbirinden tamamen bağımsız oluşturulacak ASM kodları ile çözülecektir.

a) Yanda verilen sistemde kullanıcı, Port2 ve Port3'ye bağlı butonlardan, 0 ile 15d arasında keyfi değerler girmektedir.

Port2'nin yüksek nibble değeri ile Port3'ün düşük nibble değerleri maskelenerek okunacak ve gerekli işlemler sonrası elde edilen veriler sırasıyla r0 ve r1'e yazılacaktır.

r0 ve r1'daki değerler eşit ise Led1'i yakıp söndürünüz ve programın başa dönmesini sağlayınız. Kontrol işlemi için JZ komutu kullanılacaktır.

b) Önce Btn1'e basılıp henüz çekilmeden Btn2 butonuna basılıp çekilirse, Led2'yi yakınız ve 5! sonucunu döngü ile hesaplayıp acc'ye yazınız ve tekrar başa dönünüz.

**ÇÖZÜM:**

a.

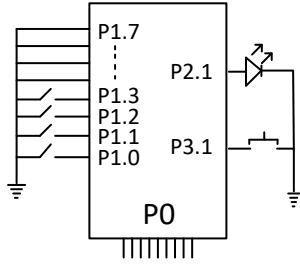
<pre> #include "at89x52.h" ORG          0000 basla:     mov      a, p2     anl      a, #0fh     mov      r0, a      mov      a, p3     swap     a     anl      a, #0fh     ; yada tam tersi ilk önce #f0h     ; ile maskele sonra swap yap     mov      r1, a      subb     a, r0 ; cikarttik     jz       x     sjmp     basla x:     setb     p0.0     sjmp     basla ; istege bagli end </pre>	<pre> #include "at89x52.h" ORG          0000 x:     jb       p1.0, y     jb       p1.1, y m:     jnb      p1.1, m     setb     p0.1     mov      r0, #5d     mov      a, #1d; carpim dgskn  carp:     mov      b, r0     mul      ab     djnz     r0, carp     ; zaten carpim acc'de y:     sjmp     x end </pre>
---	---

Başarılar dileriz...

Dr. Şuayb Çağrı YENER

Dr. Burhan BARAKLI

4)



Sifreler: db 10h,22h,38h.....,7fh  
Tabloda 16 adet deęer vardır

Ařaęıdaki iki řık birbirinin devamı olmakla birlikte her řıktaki çözüümü yalnızca istenenlere yönelik ASM kodunu verecek biçimde oluřturunuz.

- P3.1'e baęlı butona her basılıp-bırakıldığında P1 portundan kullanıcının girdięi 1 Baytlık veri (x verisi) okunacak ve r0 yazılacaktır. Ardından tablodaki x. sıradaki bilgi okunarak P0'da gösterilecektir.
- x bilgisi veri tabanında (tabloda) mevcut ise p2.1 baęlı led yakılacaktır. (x deęerinin araması/bulunması için veri tabanındaki deęerler ile gerekli karşılařtırmalar/kontroller yapılacaktır).

### çÖZÜM:

```
#include "at89x52.h"
ORG      0000
mov      dptr,#sifreler
x:       jb      p3.1,x
y:       jnb     p3.1,y
mov      r0,p1
mov      a,r0
movc     a,@a+dptr
mov      p0,a
// soru a bitti

// b soru
mov      r1,#00h
m:       mov      a,r1
movc     a,@a+dptr
cjne     a,r0,devam
sjmp     yak
devam:   inc      r1
cjne     r1,#17d,m
sjmp     x
yak:     setb     p2.1
sjmp     x ; istege bagli yada k: sjmp k yaz

sifreler: db 10h,22h,38h,33h,65h,6h,7h,9h,19h,31h,54h,53h,40h,0a5h,0bh,7fh
end
```

// sorunun b sikkı birçok deęişik yöntemle çözülebilir.