

Sıra No: (Listeden bakıp yazınız)	EEM 304 MİKROİŞLEMCİLER 2016 – 2017 Bahar Yarıyılı	
Ad:		
Soyad:	VİZE	
Öğrenci No:	31.03.2017	
İmza:	Süre: 100dk	

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

Açıklama:

- Çözüm için aşağıdaki şablon şekilleri kullanınız.
- RAM’i derste anlatıldığı biçimde en altı 00h adresi olacak biçimde planlayınız, değeri değişen/gösterilecek hücreleri şekilde adresini sağa yazmak kaydıyla uygun biçimde oluşturunuz.
- 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.
- Tüm hücrelerde değişen değerlerin üzerini rahatlıkla okunabilecek şekilde çiziniz. Üzeri çizilmeyen değer son değer olacaktır.

```
#include
<ADUC841.H>
ORG 0000H
MOV R5,#07H
MOV PSW,#88H
MOV A,#01
MOV 08H,#06H
MOV @R0,#02H
MOV 2FH,#0FFH
INC A
RLC A
MOV C,7FH
SETB RS0
SETB RS1
CPL C
MOV R0,A
MOV @R0,#0AH
CLR 78H
CLR PSW.4
PUSH 05H
PUSH 06H
POP 05H
END
```

RAM

FF-FE	2F
05	18
02	09
06-0A	08
02	06
07-0A-02	05

PSW

C				RS1	RS0				
Binary									Hex
1	0	0	0	1	0	0	0	0	88
								1	89 <small>(PLU ANL YOK)</small>
0								0	08
1									88
				1					98
0									18
0	0	0	0	1	0	0	0		08

A

01
02
05

SP

07
08
09
08

Başarılar dileriz...

2) Alttaki program kodunun bir kez döngüsünü tamamlaması (SJMP ile BASLA etiketine dönmesi) göz önüne alınarak,

Karşılaşılan her alt programa ait komutta (**ACALL, LCALL, RET**) programın **nereden nereye dallanacağını kod metni üzerinde okla gösteriniz** ve bunu sırayla (1, 2, 3,...) numaralandırınız.

Ayrıca her gösterdiğiniz dallanma için **ilgili dallanma komutunun işlenmeden hemen önceki ve sonraki anına** karşılık gelmek üzere aşağıda verilen tabloya PC, SP kaydedicileri ile belirtilen RAM hücrelerinin değerlerini yazınız.

Adres	Program Kodu
	ORG 0000H
	BASLA:
0000	MOV P0,#00H
0003	MOV R0,#20H
0005	MOV @R0,#22H
0007	SETB RS0
0009	MOV A,#04H
000B	LCALL ALT_L
000E	MOV P0,#0FH
0011	NOP
0012	SJMP BASLA
	ALT_L:
0014	SETB 00H
0016	MOV A,20H
0018	MOV R0,A
0019	ACALL ALT_A
001B	MOV 0AH,#0CH
001E	NOP
001F	RET
	ALT_A:
0020	MOV 06H,#11H
0023	NOP
0024	RET
	END

Dallanma	Komut	Durum	RAM adresleri							
			PC	SP	06	07	08	09	0A	0B
1	LCALL	Önce	0B	07	00	00	00	00	00	00
		Sonrası	14	09	00	00	0B	00	00	00
2	ACALL	Önce	19	09	00	00	23	00	00	00
		Sonrası	20	0B	00	00	23	00	1B	00
3	RET (ALT_A)	Önce	24	0B	11	00	23	00	1B	00
		Sonrası	1B	09	11	00	23	00	1B	00
4	RET (ALT_L)	Önce	1F	09	11	00	23	00	0C	00
		Sonrası	23	07	11	00	23	00	0C	00
5	RET (ALT_A)	Önce	24	07	11	00	23	00	0C	00
		Sonrası	11	05	11	00	23	00	0C	00

Program 1 kez tamamlanıp BASLA etiketine dallanıldığında aşağıdaki kaydedicilerin son değerleri:

SP

05

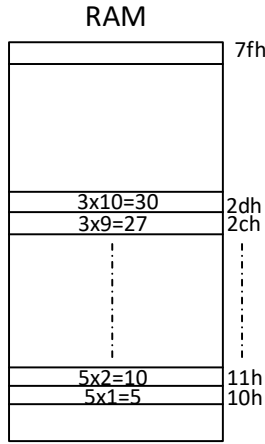
P0

00

Sıra No: (Listeden bakıp yazınız)	EEM 304 MİKROİŞLEMCİLER 2016 – 2017 Bahar Yarıyılı	
Ad:		
Soyad:	VİZE	
Öğrenci No:	31.03.2017	
İmza:	Süre: 100dk	

3)

RAM'i sıfırlayan asm kodunu yazınız ve yukarıda verilen çarpım tablosundaki 3'lük, 4'lük ve 5'lik çarpım tablosu sonuçlarını, gösterilen ilgili adreslere yazan programı, iç içe döngü mantığı ile kodlayınız



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

ÇÖZÜM:

```
#include "aduc841.h"
org 0000
; ram sıfırlama
mov     r0,#7fh

x:
mov     @r0,#00h
djnz   r0,x

;-----
; carpim tablosu

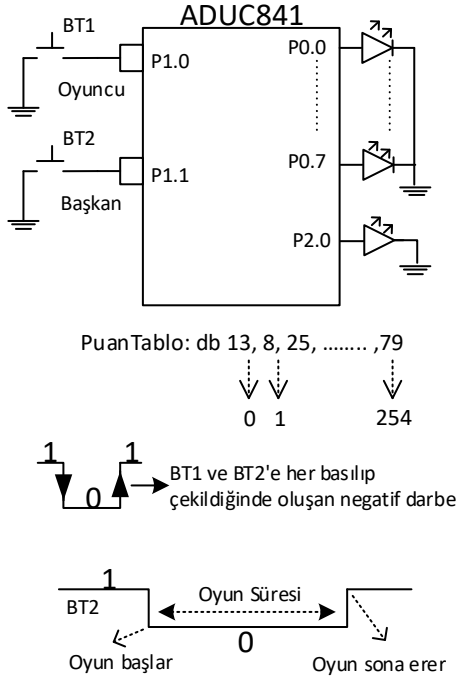
mov     r0,#10h ; adres
mov     r2,#5d ; j

; UCLUK DONGU 5-4-3
m:
; onluk dongu
mov     r1,#1d ; 10 luk dongu i
z:
mov     a,r1
mov     b,r2
mul     ab
mov     @r0,a
inc     r0
inc     r1
cjne   r1,#11d,z
; onluk dongu bitti
dec     r2
cjne   r2,#2d,m
;-----
bekle: sjmp     bekle
```

Başarılar dileriz...

end

4)



Yandaki şekilden faydalanarak bir oyun tasarlanmak istenmektedir. Oyun **başkan**'ın BT2 butonuna bastığı anda **başlayacak** ve **basık tuttuğu** sürece oyun **devam edecektir**.

Oyun başladığı anda **oyuncu**, oyun süresi boyunca olabildiğince BT1 butonuna basıp-çekmeye başlayacak ve basım sayısı RAM'in 10h adresinde tutulacaktır.

- Oyun sonlandıktan sonra** (Başkan, BT2 butonunu ilk durumuna alınca) p2.0 bağlı led yakılacak ve kişinin kazandığı puan miktarı basım sayısından tablo değerine göre **PuanTablo**'sundan okunarak kazandığı puan miktarı p0'da gösterilecek ve oyun tekrar baştan başlatılacaktır.
- Oyun devam ederken** (oyun süresi boyunca), basım sayısı 255 sayısına ulaşmış ise oyundan çıkılacak, p0'daki ledlerin hepsinin yanması sağlanacak ve sonsuz döngüde beklenecektir.

ÇÖZÜM:

```
#include "aduc841.h"
org 0000
```

basla:

```
mov    p0,#00h
mov    10h,#00h
clr    p2.0
mov    dptr,#kazancdegeri
```

```
;-----
x:jb   p1.1, x      ; oyunun başlamasi bekleniyor
```

```
oyun:      ; oyun basladi
```

```
first:  jb   p1.0,first ; butonun basılıp çekilmesi
second:  jnb  p1.0,second
```

```
inc    10h
mov    a,#255d
subb   a,10h
jz     oyunbitti2
```

```
jnb   p1.1,oyun      ; oyun devam ediyormu??
```

```
;-----
```

```
; oyun bitti1
mov    a,10h
movc   a,@a+dptr
mov    p0,a
sjmp   basla
```

oyunbitti2:

```
mov    p0,#255d
x1:    sjmp  x1
```

kazancdegeri: db 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 ; 256 tane deger olacak random degerler

end