

1. List the sequence of microoperations for executing $M[ADRS] \leftarrow M[ADRS] + AC$ instruction.
 $M[ADRS] \leftarrow M[ADRS] + AC$ emrinin çalıştırılması için gereken mikrooperasyonlar dizisini listeleyiniz.

$DR \leftarrow M[PC]$
 $IR \leftarrow DR$
 $PC \leftarrow PC + 1$
 $DR \leftarrow M[PC]$
 $AR \leftarrow DR$
 $PC \leftarrow PC + 1$
 $DR \leftarrow M[AR]$
 $AC \leftarrow AC + DR$
 $DR \leftarrow AC$
 $M[AR] \leftarrow DR$
 $TC \leftarrow \emptyset$

2. List the advantages of having many addressing modes in addition to direct addressing mode. Doğrudan adresleme moduna ek olarak birçok adresleme modunun avantajlarını listeleyiniz.

1. Programcıya, yüksek seviyeli dillerdekine benzer, gösterici kullanımı, döngü sayacıları, veri indisleme ve program kaydırma gibi çok yönlülük sağlama,
2. Emirlerin adres kısmında kullanılan bit sayısının azaltılması

3. Compute the effective address and indicate accumulator content for the **Load to AC** accumulator instruction for direct, immediate, indirect, relative, index, register and register indirect addressing modes for the register and memory contents given below. Aşağıda verilen kaydedici ve bellek içerikleri için, akümülatöre yükle emrinin doğrudan, emiri izleyen, dolaylı, göreceli, indisli, kaydedici ve kaydedici üzerinden dolaylı adresleme modları için geçerli adres değerini hesaplayınız ve akümülatör içeriğini gösteriniz.

PC R1 AC

Memory	
0243	LD AC Mode
0244	0507
0245	ST AC Mode
0411	1234
0507	0845
0752	2345
0845	3456
0918	4567

Mode	Register Transfer	Eff. Adr.	AC
Direct	$AC \leftarrow M[ADRS]$	0507	0845
Immediate	$AC \leftarrow NBR$	0244	0507
Indirect	$AC \leftarrow M[M[ADRS]]$	0845	3456
Relative	$AC \leftarrow M[ADRS + PC]$	0752	2345
Indexed	$AC \leftarrow M[ADRS + R1]$	0918	4567
Register	$AC \leftarrow R1$	—	0411
Regist. Ind.	$AC \leftarrow M[R1]$	0411	1234

4. The **interrupt** method to service requests is in principle similar to a **subroutine call** to handle some task. List the differences between the two. **Kesme** metodu ile isteklerin servis görmesi prensip olarak bir işi yapmak için **altprogram çağırmaya** çok benzemektedir. Bu ikisi arasındaki farklılıkları listeleyiniz.

1. Altprogramlar, kullanıcının bir emri ile çağılırken, kesmeler dış veya iç işaretler ile başlatılır.
2. Kesme servis altprogramının adresi, altprogramda olduğu gibi emrin adres alanından alınmaz, bir donanım ile belirlenir.
3. Altprogram çağrılarında yapıldığı gibi yalnız geri dönüş adresi değil, işlemcinin o andaki tüm kaydedici içerikleri saklanmalıdır.

5. The x86 Assembly code snippet given below is compiled by debug. Write necessary debug commands to store the code in a file named `control.com`. Aşağıda verilen x86 Assembly dilindeki kod parçası debug ile Assembly diline derlenmiştir. Kodun `control.com` adlı bir dosya içine saklanması için gereken debug komutlarını yazınız.

17C2:0100	8B164B3A	MOV	DX, [3A4B]
17C2:0104	ED	IN	AX, DX
17C2:0105	3B064D3A	CMP	AX, [3A4D]
17C2:0109	7F02	JG	010D
17C2:010B	EBF7	JMP	0104
17C2:010D	8B164F3A	MOV	DX, [3A4F]
17C2:0111	A1513A	MOV	AX, [3A51]
17C2:0114	EF	OUT	DX, AX

```
-r cx
cx:0000
15
-n control.com
-w
-q
```