

1- Verilen program parçasında NUM = F0E0 olduğuna göre programın çalışması sonucu Count'ın içeriğini bulunuz. Count =? **(10 puan)**

	MOV	CX,16
	SUB	BX,BX
	MOV	AX,NUM
Back:	SHR	AX,1
	JC	SON
	INC	BX
Son:	LOOP	Back
	MOV	Count,BX

2- Verilen program parçasının çalışması sonucu CX registeri'nin içereğini yazınız. **(11 puan)**

```

MOV      CX, 12
TOP: DEC  CX
      MOV  AL, CL
      MUL  AL
      CMP  AL, 121
      LOOPZ TOP

```

3-Verilen program parçası içerisinde ADD komutunun kaç kere kullanıldığını yazınız. **(10 puan)**

```

MOV      CX, 8
TOP: ADD  BL, [SI]
      INC  SI
      DEC  CX
      LOOP TOP

```

4-Belleğin DS:1000 DS:100F arasında bulunan 16 adet birer byte'lık veriye 5 sayısını ilave eden bir program parçası yazınız. Program içerisinde LOOP komutu kullanılacaktır. **(13 puan)**

5- Eğer AX = FFFF ve BX = FFFF ise ADD AX,BX işlemin sonucuna göre Flag'lerin durumunu yazınız. **(10P)**

SF	CF	ZF	PF	OF	IF	TF

6 a) Flag register de bulunan 3 kontrol falg'ın adını yazınız b) Call komutu ile alt program çağırılırken hangi veriler yığında (stack) saklanır. **(6P)**

7 -Verilen programın sonucunu (çıktısını) registerlere bağlı bir denklem şeklinde ifade ediniz.(örneğin..7AX-15CX..) **(12 puan)**

```

PUSH AX
XOR AX,AX
MOV AX,4
MUL BX
ADD AX,1
PUSH AX
MOV AX,CX
MOV CX,6
AND DX,00
DIV CX
POP BX
SUB AX,BX
POP BX
SUB DX,DX
DIV BX

```

DATA1	DB	+80
DATA2	DB	+85
sunuc	DW	?
	MOV	AL,DATA1
	MOV	BL,DATA2
	ADD	AL,BL ; OF =...;
		CF=...;SF=...AL=....
	JNO	HIGH
	MOV	AL,DATA2
	CBW	;
	MOV	BX,AX
	MOV	AL,DATA1
	CBW	
	ADD	AX,BX; OF =...;
		CF=...;SF=...AX=....
HIGH	MOV	SONUC,AX
		<i>Soru 8' in programı</i>

8- Verilen programa çalışmasına göre her aşamada verilen boşlukları doldurunuz.

(10 puan)

9- Verilen program parçasının çalışması sonucu BL registeri'nin içereğini yazınız

AL= 10101011

(10 puan)

```

MOV CX,6
SUB BL,BL
L1:  SHR AL,1
      ADC BL,0
      LOOP L1

```

10- AX =000F, BX= 003F, CX = 00AC, DX = 00B1 olduğuna göre programın çalıştırılması sonucu AX registerin içereğini yazınız.

(8 puan)

```

CMP AX,BX
JC T2
MOV AX,BX
T2:  CMP AX,CX
JC T3
MOV AX,CX
T3:  CMP AX,DX
JC DONE
MOV AX,DX
DONE: . . . .

```