

**Bilgisayar Mühendisliği Bölümü**  
**BLM2021 - Alt Seviye Programlama**  
**Bütünleme Sınavı**

Grup No :  
Numara :  
Ad Soyad:  
İmza :

05/02/2021  
100 Dakika

**Sınav Kuralları:**

- Yazı ve çizimlerinizin **okunaklı** olması menfaatiniz gereğidir.
- **Kopya** çekmek/vermek en ağır şekilde **cezalandırılacaktır**.
- Cevaplarınızı **word dosyası olarak online.yildiz.edu.tr'ye yükleyiniz**.

1.soru (24)	2.soru (16)	3.soru (30)	4.soru(25)	5.soru(25)	Toplam(120)

Başarılar

**S O R U L A R**

1. Aşağıdaki ifadelerin başına **D (doğru)** veya **Y (yanlış)** olduklarını işaretleyiniz. **(12x%2)**

a)	Alt yordam, başka bir yordam tarafından kullanılacak ise, tanımlı olduğu dosyada EXTRN tanımlanmalıdır.
b)	Aynı fiziki adres değeri, farklı kesim ve işaret yazmaç ikilileri ile oluşturulabilir.
c)	Ana yordamın yığın üzerinden aktardığı 1 adet Word değeri, alt yordam tarafından POP AX ile yığından kaldırabilir.
d)	8086 işlemcilerde kesme sadece dış kaynaklı (INTR ve NMI uçları ile) oluşturulabilir.
e)	Kullanıcı kesme yordamlarının sonuna/arkasına kendi yazdığı kodu ekleyerek, her kesme oluştuğunda kendi konun çalışmasını sağlayabilir.
f)	DOS kesmesi içinden BIOS kesmesi çağırılabilir
g)	Vektör tablosu, kesme yordamlarının yer aldığı tablodur.
h)	Ana yordamın kullandığı macro kodu, aynı kod kesimi içinde herhangi bir yerde tanımlanabilir.
i)	Ana yordamın çağırdığı NEAR yordam kodu, aynı kod kesimi içinde herhangi bir yerde tanımlanabilir.
j)	FAR ve NEAR tipinde CALL opcode boyutları aynıdır.
k)	Alt yordam, tanımı farklı bir dosyadaysa, ana yordamda EXTRN olarak tanımlanmalıdır.
l)	DOS kesmesi içinden DOS kesmesi çağırılabilir.

2. Aşağıdaki soruda boşlukları uygun şekilde doldurunuz. **(8x%2)**

- a) Vektör tablosunun büyüklüğü \_\_\_\_\_ byte'tır.
- b) Kesme, eğer \_\_\_\_\_ geliyor ise maskelenebilir.
- c) COM tipi programda veri tanımı \_\_\_\_\_ yapılacak olursa koda yapılacak eklemeler veri adreslerini değiştirmez.
- d) 8086 mikroişlemcisi \_\_\_\_\_ bit adres yolu genişliğine ve \_\_\_\_\_ bit veri yolu genişliğine sahiptir.
- e) 8086 gerçek kipte çalışırken CS = 0FFFFH ve IP = 0111H ile erişilen fiziki adres \_\_\_\_\_ olur.
- f) IRET komutu çalıştırıldığında SP değeri \_\_\_\_\_ artar.
- g) 8086 işlemcilerde 255 numaralı kesme çağırıldığında IP [\_\_\_\_\_]H ve CS [\_\_\_\_\_]H adreslerinden yüklenir.

h)	MOV AX, 1 MOV BL, -1 MUL BL	işlemleri sonucunda AX yazmacı değerleri _____ olur.
----	-----------------------------------	--

3. Üç boyutlu uzayda **vektörel çarpım işlemi** herhangi  $A=(a,b,c)$  ve  $B=(u,v,w)$  vektörleri için aşağıdaki şekilde tanımlanır.

$$A \times B = (bw - cv, cu - aw, av - bu)$$

Bir girdi vektörü, herbiri işaretli, byte büyüklüğünde 3 adet sayıdan oluşur. Buna göre A ve B vektörlerini **yığın üzerinden alan**, verilen formüle göre vektörel çarpımını hesaplayan ve **sonucu yine yığın üzerinden döndüren** **VECTOR\_MUL** isimli **harici yordamını** yazınız. (%30)

NOT: Yordamı yazarken kesinlikle veri alanı tanımlı yapmayınız.

4. 900 elemanlı, byte büyüklüğünde, işaretli elemanları olan DIZIA ve DIZIB isimli vektör dizilerinin karşılıklı (aynı indisli) elemanları için vektörel çarpımı **VECTOR\_MUL** isimli **harici yordam** yardımıyla hesaplayıp, işlem sonucunda dönen değerlerin kaç tanesinin herhangi bir eksen üzerinde olduğunu bulan ve **OVERXYZ** isimli **değişkende saklayan** assembly programı yazınız. (%25)

NOT: Soru 3 de tarif edildiği gibi **VECTOR\_MUL** isimli **harici yordamı**, **parametreleri yığın üzerinden alır ve sonucu yığın üzerinden döndürür**. Vektör değeri dizilerin birbirini takip eden üç elemanıdır. Dönen değerlerden **en az biri 0 (sıfır) ise eksen üzerinde olduğu** anlaşılır.

5. a) Aşağıdaki **veri tanımlamaları** yapılp, **debug** ortamında veri kesimine bakıldığında, **bırakılan boşluklarda** neler görünecektir? Yazınız (%10)

```
mydata    segment para 'data'
           dw 1,14,4,7,9,-3,2,3,16,18h,0ah
mydata    ends
```

```
DS:0000    _ _ _ _ _ _ _ _ _
DS: _ _ _ _ _ _ _ _ _
DS: _ _ _ _ _ _ _ _ _
```

b-) Yukarıdaki **veri kesimindeki değerler** ile aşağıdaki **kod parçası sonuna kadar çalıştırılır ve veri kesimine debug ortamında** bakılırsa ne görüneceğini yazınız (%15)

```
-u
0D04:0009 A11400      MOV     AX,[0014]
0D04:000C 8D360000    LEA     SI,[0000]
0D04:0010 48          DEC     AX
0D04:0011 D1E0      SHL     AX,1
0D04:0013 8BFE      MOV     DI,SI
0D04:0015 03F8      ADD     DI,AX
0D04:0017 3BF7      CMP     SI,DI
0D04:0019 7D1F      JGE     003A
0D04:001B F7040100    TEST    WORD PTR [SI],0001
0D04:001F 7414      JZ      0035
0D04:0021 F7050100    TEST    WORD PTR [DI],0001
0D04:0025 7509      JNZ     0030
0D04:0027 8B04      MOV     AX,[SI]
0D04:0029 8705      XCHG    AX,[DI]
0D04:002B 8904      MOV     [SI],AX
0D04:002D 83C602      ADD     SI,+02
0D04:0030 83EF02      SUB     DI,+02
0D04:0033 EBE2      JMP     0017
0D04:0035 83C602      ADD     SI,+02
0D04:0038 EBDD      JMP     0017
0D04:003A CB      RETF
```

```
DS:0000    _ _ _ _ _ _ _ _ _
DS: _ _ _ _ _ _ _ _ _
DS: _ _ _ _ _ _ _ _ _
```