

Sınav Bilgisi	EEM-364 Mikro İşlemciler FİNAL SINAVI B... Grubu (Numara Sonu Tek İse: A) (Numara Sonu Çift İse: B)	05/06/23 11:30	1	2	3	4	5	6	7	8	9	10	11
Öğrenci İsmi	[Redacted]	Numarası	[Redacted]										
Breakout Odası	[Redacted]	Toplam Kâğıt		S1	S2	S3	S4	S5	S6				
İmza	[Redacted]	Süre 90 dk	Kopya çektiğinden şüphelenilen ya da kurallara uymayan öğrenci hakkında tutanak tutulacaktır, sınavının geçersiz sayılıp sayılmamasına ve/veya örgün sınava alınıp alınmamasına kurul karar verecektir. Bilginizin değerini usulsüz paylaşımlarla düşürmeyiniz.										

(DOĞRU GRUP (A ve B) SEÇİM YAPILMAYAN SINAV KAĞIDINDAN 20 PUAN KIRILACAKTIR)

HEX Sayılar					
A	B	C	D	E	F
10	11	12	13	14	15

Bit Değerleri			
8	4	2	1

Dallanma Komutları		Kesme Değerleri	
JZ: Sıfır	JLE: küçük ve eşit	01H	STDIN'den karakter oku
JNZ: Sınır değil	JE: eşit	02H	STDOUT'a karakter yaz
JMP: Koşulsuz	JNS: işaretli değil	09H	STDOUT'a dizi-metin yaz
JG: Büyük	CMP: Karşılaştır	4CH	Programı sonlandır
JL: Küçük		Diğer	
JGE: Büyük ve eşit		include "emu8086.inc" define_print_num_uns call print_num_uns	
Makrolar			
putc: karakter koyma gotoxy: imleç konumlandırma			
Başarılar Dilerim...			

S1A) Makro ve Prosedür arasındaki farkı açıklayınız. Bildiğiniz iki makroyu if.

- Makrolar ana fonksiyon içerisinde gerçekleştirilen, prosedürler ana bölgenin dışında geliştirilir ve gerçekleştirilir. Prosedürler "call" ile çağırılıp geliştirilir. Makrolar için böyle bir işlemler yoktur.

- "putc" makrosu ekrana karakter basmaya yarar. Yani "putc "a" ifadesi ekrana bir tane "a" işareti koyar. "gotoxy" makrosu ekran çıktısında ekrana basılacak ifadenin konumunu belirtir. Yani imlecin konumunu ayarlamaya yarar. "gotoxy ch, cl" şeklinde kullanılır. "ch" kolonu ifade eder, "cl" satırları ifade eder.

S1B) "sembol" kavramını ve değişkenden farkını kısaca açıklayınız.

Diğer programlar dillerinde ki değişken kavramı, assembler programlama dilinde "sembol" olarak geçer farkı ise değişkenler, ona atanan değer'in kendisini tutarken, "sembol"ler ona atanan değer'in adresini tutar.

S1C) Aşağıda verilen tanımların doğru ve yanlış olduklarını işaretleyiniz. Yanlış ise doğrusunu ifade ediniz.

- 1: D
- 2: D
- 3: Y → BH yazmacında değil DH yazmacında tutulur.
- 4: D
- 5: Y → Stack Segmenti 16 bit değil 8bit üzerinden çalışır.
- 6: Y → AX indisi yazmacı olarak kullanılamaz.
- 7: Y → gotoxy bir prosedür değildir, makrodur.

S2A) Aşağıdaki kod parçasının ne görev yaptığını açıklayınız. İşlem sonunda A ve B değerleri ne olurdu? (B grubu)

- "Y" dizisindeki elementlerin "L:" de çift mi tek mi olduğu kontrol edilmiş. Eğer çift ise "A" sembolünde değerler toplanmış. Tek ise "B" sembolünde değerler toplanmış.
- "D:" ise dizinin elementlerinde dolusmak için inc yapılmış ve dizinin sonuna gelinip gelinmediği kontrol edilmiş.
- işlem sonunda : A'nın değeri: 14 olur.
B'nin değeri: 12 olur.

- ilk önce "Y" dizisinin ilk elemanı "al" yazmacına alınır. sonra "test" ile T16 kontrolü yapılır. "G" ise "ZF" ye dayanır, "T" ise "Y4" ye dayanır. Sonra duruma göre A veya B ile "al" yazmacı toplanır.

- S2B) Aşağıda verilen kodda 5 tane hatayı tespit edip açıklayın.
- 06. satırda "lx db #x" yapılmış. "#" yerine "\$" işareti konulmalıydı.
 - 51. satırda "main proc far", sadece "endp" ile kapatmış. "main endp" olması gerekiyordu.
 - 21. satırda "mov cx, lx" atanması hatalıdır "lx" 8 bit'ken cx 16 bit'tir. "lx equ \$-x" olarak tanımlanseydi doğru olurdu.
 - 28. ve 29. satırlarda dallenacak satırlar yanlış girilmiş "zz" yerine "zz" olmalıydı, "yy" yerine "yy" olmalıydı.
 - 26. satırda 8-16 bit hatası var. ve ayrıca "-Ax" yazması "-" olamaz.
 - 35. satırda "k" 1'ye dallenacak bir yer yok. "k" yerine "K" olmalıydı.

S3A) Aşağıdaki işlemlerin sonucunda AX ve DX yazmaçlarının içi nedir.

İşlem 1:

AX	
AH	AL
02	03

BL	
14	3

MUL BL →

AX	
AH	AL
00	3C

BL 14
AL 3
3C4
→ AX'e yazılır

→ AH'inde de "0" yazılır.

İşlem 1 için "AX: 003CH" olur
DX: Değişmez.

İşlem 2:

AX	
AH	AL
00	33

BX	
BH	BL
00	09

DIV BX →

DX	
DH	DL
06	03

AX 33
BX 9
27 3
6 → kalan

→ kalan 6, bölüm 3

İşlem 2 için "AX: 0033H" olur
"DX: 0603H" olur.

S3B) "SHL AL, 1" komutu AL yazmasını 1 sola kaydırır.
12x(10) işlemini sadece SHL ve ADD komutu kullanarak yapınız.

- MODEL SMALL
- DATA
- CODE

```
Add AL, 12
SHL AL, 10
HLT
```

S3C) Sadece kodu verilen programı yazınız.

- Dışarıdan bir sayı al (A)
 - Eğer $5 < A < 8$ ise Ekrana "D" yazdır. Değilse "Y" yazdır.
- ```
include "emu8086.inc"
• MODEL SMALL
```

• DATA

A db 0

• CODE

```
mov ax, @data
mov ds, ax
```

```
mov AX, 0
```

```
mov AH, 02H
int 21H
```

```
cmp AL, "5"
```

```
jge D
```

SON:

```
HLT
```

D: cmp AL, "8"

```
jg S
```

```
putc "D"
```

```
jmp SON
```

S: putc "Y"

```
jmp SON
```

S4A) Dışarıdan 10 tane karakter alınız. Aşağıdaki ekranı üretecek olan kodu yazınız.

- Arada "-" olacak
- Dizi tersten bastırılacak
- İlk 3 karakter açık dişerleri "\*" olacak şekilde basılacak.

ör  
123456789A -

```
include "emu8086.inc"
```

```
.MODEL SMALL
```

```
.DATA
```

```
x db 10 DUP(0)
```

```
.CODE
```

```
mov ax, @data
```

```
mov ds, ax
```

```
mov ax, 0
```

```
mov si, 0
```

```
Li mov ah, 01H
```

```
int 21H
```

```
mov x[si], AL
```

Berhan Berk Akşın / 20120205310 / ~~1~~ P

548) Aşağıdaki programda verilen X sembolü altında  $X \in \mathbb{R}^{4 \times 4}$  bir matristir. Matristeki değerler eşer bulundukları satırdan büyük ise V sembolünde toplanacaktır. Bu programın etkisini siziniz, programı yazınız.

| ör/ | <u>satır</u> | <u>Matris</u> | <u>Toplam Değer</u> |
|-----|--------------|---------------|---------------------|
| 0   |              | 2, 0, 1, 4    | 7                   |
| 1   |              | 1, 0, 2, 3    | 5                   |
| 2   |              | 3, 2, 1, 0    | 3                   |
| 3   |              | 4, 1, 1, 5    | 9                   |