

No:

Şube No:

18.11.2016

Ad Soyad:

90 dk.

Pamukkale Üniversitesi Bilgisayar Mühendisliği
CENG 103 Bilgisayar Mühendisliğine Giriş Ara Sınavı

Soru	1 (20p)	2 (10p)	3 (10p)	4 (15p)	5(15p)	6(15p)	7(25p)
Puan							

1. Bu soru için son sayfadaki LDT'yi inceleyiniz. PC=0 ile makine çalışmaya başladığına göre

a. Makine sonlanana kadar kaç komut çalıştırılır?

_____4_____

b. Makine sonlandığında PC'nin içeriği ne olur?

_____09_____

Adres	İçerik	Adres	İçerik
00	B0	07	C0
01	03	08	00
02	25	09	23
03	B0	0A	B0
04	0C	0B	03
05	C0	0C	B0
06	00	0D	07

2. Aşağıdaki görevleri LDTye göre kodlayınız.

a. Register 5 ve register 6'yı EXOR'la sonucu register 2'ye yaz 9256

b. Register 3'ün içeriği register 0'ın içeriğine eşit ise 77 adresindeki komutu çalıştır.
B377

3. Aşağıdaki değerleri 8 bitlik floating-point notasyonuna göre yazınız. Yuvarlama hatası olanları vurgulayınız.

a. $2 \frac{3}{4}$

b. $5 \frac{1}{4}$

c. $\frac{3}{4}$

d. $-(3\frac{1}{2})$

e. $-(4\frac{3}{8})$

a. 10.11 (01101011) b. 101.01 (01111010) hata c. 0.11 (01001100)

d. - 11.1 (11101110) e. - 100.011 (11111000) hata

4. Aşağıdaki tabloda verilen error correction code'unu (hata düzeltme kodunu) kullanarak aşağıdaki mesajları çözünüz.

Sembo l	Kodu
B	0000000
D	0001111
E	0010011
G	0011100
İ	0100110
M	0101001
R	0110101
S	0111010
V	0101111
Z	1110000

- a. 0000000 0101000 0011101 b. 0001110 0010011 0110001 0111010 1100110
 B M G D E R S İ
- c. 1101111 1100110 1110000 0010011 0111010 0100110
 V İ Z E S İ

5. Aşağıdaki boş yerlere listedeki uygun terimleri yerleştiriniz.

shell (komut satırı), CSMA/CD, graphical user interface (GUI, grafiksel arayüz), virtual memory (sanal bellek), boot loader, Domain Name System, process, deadlock (ölü-kilit), fetch

...**Virtual memory**..... ikincil belleğin ana bellekmiş (main memory) gibi kullanılmasını sağlar.

.....**Process**..... programın CPU'da (işlemci) çalışır halidir.

.....**Boot loader**..... sayesinde işletim sistemi ikincil bellekten birincil belleğe yüklenir.

...**Deadlock**..... birbirlerinin aldığı kaynakları bekleyen processlerin birbirlerini sürekli beklemesi ve iş yapamaması durumudur.

...**CSMA/CD**..... bus topolojilerinde kullanılan, aynı hattı kullanan bilgisayarlar arasında hattın aynı anda kullanılmasından kaynaklanan sorunları azaltmaya çalışan bir protokoldür.

.....**shell**..... üzerine çeşitli komutlar yazılarak işletim sisteminin sağladığı fonksiyonlardan yararlanılabilir.

.....**GUI**..... işletim sisteminin sunduğu olanakları kullanabildiğimiz görsel bir araçtır.

.....**Fetch**..... makina cycle'ının (döngüsünün) ilk adımıdır.

.....**DNS**..... kullanılarak makinaların alan adları (akılda kalıcı isimleri) ip adreslerine dönüştürülür.

6. 20'den 28'e kadar ana bellekte aşağıdaki bit desenleri bulunmaktadır.

<u>adres</u>	<u>içerik</u>
20	12
21	20
22	32
23	30
24	B0
25	21
26	24
27	C0
28	00

Makina PC=20 ile başlamaktadır. Buna göre program sonlandığında;

- 0,1, ve 2 registerlarında (yazmaç) hangi değerler bulunur?
- Ana belleğin 30 adresinde hangi değer bulunur?
- Ana belleğin B0 adresinde hangi değer bulunur?
 - Registers 0, 1, and 2 will contain 32, 24, and 12, respectively.
 - 12
 - 32

7. Bazen probleme uygulanan küçük bir değişiklik çözüm yolunu değiştirebilmektedir.

İlk problemimiz için bir grup insanı iki farklı gruba ayırmalısınız öyle ki iki farklı grubun yaşlarının kendi içindeki yaş toplamının farkı olabilecek en büyük değeri alacaktır (İki grubun da yaş toplamı bulunacak, büyük olan küçük olandan çıkarılacaktır.). Bunun için toplam insan sayısının ve bu insanların isim ve yaşlarının bir liste olarak verildiğini, sonuçta oluşan gruptaki insanların isim listesini ve yaş farkını bulan bir algoritma yazınız.

Şimdi problemi değiştirelim ve iki grubun yaş toplamı farkının minimum olacağı iki grubu bulan bir algoritma yazınız.

Not: Grupların eleman sayısı istediğiniz gibi olabilir.

To solve the first problem, sort the group by age, and form the subgroups by placing the youngest person in one while putting all the rest in the other

Def findMax(list,size):

```
group2={}
list=Sort(list)
max=0
group1=list[0]
group2= list[1-to-size]

sum1=list[0]
i=0
sum2=0
while (i<sizeof(group2)):

    sum2=sum2+group2[i]
end while
max=sum2- sum1
return max
```

group1={}

Problem2

Iteratively

def findMin(list,size):

```

minDiff=maxINT
generate all possible groups in array1 and array2 as Matrices
size=size of array2
    result1, result2 as vector
i=0

```

```

while(i< size of array1):
    j=0
    while(j< size of array1[j]):
        sum1=sum of array1[i]
        sum2= sum of array2[i]
        dif=abs(sum2-sum1);
        if dif<minDiff:
            minDiff=max;
            result1=array1[i]
            result2 =array2[i]
        end if

```

end while

```

print result1, result2 and minDiff;

```

```

min = sum;

```

```

globalVec = baseVec;

```

OR Recursively

```

fun generate(baseVec, generatedVec, position, total)
    if (abs(sum - 2*total) < min){ // check if the distribution is
better
        min = abs(sum - 2*total);
        globalVec = generatedVec;
    }
    if (position >= baseVec.length()) return;
    else{
        // either consider elem at position in first group:
        generate(baseVec,generatedVec.pushback(baseV
ec[position]), position + 1,
total+baseVec[position]);
        // or consider elem at position is second group:
        generate(baseVec,generatedVec, position + 1,
total);
    }

with generate(baseVec,"",0,0)

```

Makina Dili (LDT)

<u>OpCode</u>	<u>Operand</u>	<u>Açıklama</u>
1	RXY	R saklayıcısına (register,yazmaç) XY bellek hücresinde bulunan içeriği YÜKLE (LOAD)
2	RXY	R saklayıcısına XY 'yi YÜKLE (LOAD)
3	RXY	R saklayıcısının içeriğini XY bellek hücresine DEPOLA (STORE)
4	ORS	R saklayıcısının bit deseninin içeriğini S saklayıcısının bit desenine kopyala
5	RST	S saklayıcısının içeriği ile T saklayıcısının içeriğini topla sonucu R saklayıcısına yaz (İkiye tümleyene göre)
6	RST	S ve T saklayıcılarının bit desenlerinin kayan noktalı (floating-point) notasyona göre topla, sonucu R saklayıcısına yaz.
7	RST	S ve T saklayıcılarının bit desenlerini OR (OR kapısı)'la sonucunu R saklayıcısına yaz
8	RST	S ve T saklayıcılarının bit desenlerini AND (AND kapısı)'le sonucunu R saklayıcısına yaz
9	RST	S ve T saklayıcılarının bit desenlerini EXOR (EXOR kapısı)'la sonucunu R saklayıcısına yaz
A	R0X	R saklayıcısının içeriğini X kere sağa ROTATE et(dairesel olarak) (sonuç yine R'de)
B	RXY	R saklayıcısının içeriğini 0 saklayıcısının içeriğine eşit ise bir sonraki komut için XY adresine git değilse normal akıştaki komutu çalıştır.
C	000	HALT makine durduran komut