2008 BAHAR VİZE BİLGİSAYAR ORGANİZASYONU SORULARI  
S1.)12.125+48 isleminin sonucunu hesaplayınız.  
  
  
S2.)İki tane 8 bitlik sayı karsılastırınca bitler farklıysa 1 aynıysa sıfır veriyor.Ornegin  
01100011—01011010 ile isleme tutuldugunda 00111001 seklinde icierindekileri 1’eri sayan Sal kodunu yazınız..  
  
.data  
maske: .word 0x80000000  
x: .word  
y: .word  
z: .word  
t: .word  
sonuc: .word 0  
  
.text  
xor z, x, y  
begz z, git  
geri: bgtz z, label  
add sonuc, sonuc, 1  
label: sll z, z, 1  
b geri  
git: put sonuc  
  
  
S3.)word tipinde tanımlanmıs 50 elemanlı bir dizinin en buyuk elemanını $R5 registerına  
En kucuk elemanını $R6 regısterına yazan Mal kodunu yazınız.  
.data  
dizi: .word 0:50  
  
.text  
li $1, 50  
la $2, dizi  
lw $R4, 0 ( $2 )  
mv $R5, R4  
geri: add $1, $1, -1  
beg $1, $0, son  
add $2, $2, 4  
lw $3, 0 ( $2 )  
blt $3, $4, git1  
bgt $3, $5, git2  
b geri:  
git1: add $R4, $R3, $0  
b geri  
git2: add $5, $3, $0  
b geri  
son:  
  
  
ikinci yöntem  
  
  
.data  
ar: .word 0:50  
.text  
la $14, ar  
loop: li $8, 1  
li $9, 0  
for: add $10, $14, $9  
lw $11, ( $10 )  
lw $12, 4( $10 )  
sub $13, $11, $12  
blez $13, noswap  
li $8, 0  
sw $11, 4( $10 )  
sw $12, ( $10 )  
noswap: add $9, $9, 4  
sub $13, $9, 196  
blz $13, for  
beq $8, $0, loop  
done  
  
  
S4.)a)Bellek darbogazı nedemektir??Bunu onlemek icin hangi donanımsal ve yazılımsal islemler kullanılır??  
  
b)Col Major ile Row Majordan bahsediniz. Bunları kullanan dillere ornekler veriniz.  
  
c)Sabit uzunluklu komut nedemektir?Avantajı ve dejavantajı nedir??  
  
d)Dallanma komutları performans acısından degerlendiriniz.avantaj devantaj??

**[Bellek Darboğazı](http://cengaverhero.blogspot.com/2009/06/bellek-darbogaz.html)**

Bilgisayar içerisinde bir bileşen veya veriyolu kapasitesi üzerinde kullanılmak istenirse darboğaz haline gelir.Buna bellek darboğazı denir.Bu durumda verilere cevap süresi uzar.Diğer bileşenler darboğaz haline geçen bileşeni beklemek zorunda kalır.

[**RISC İLE CISC MİMARİLERİNİN KARŞILAŞTIRILMASI**](http://cengaverhero.blogspot.com/2009/06/risc-ile-cisc-mimarilerinin.html)

1-)Risc de load ve store komutu var Cisc de bunlara ilaveten komutlar var.

2-)Risc’de komutlar sabit 32 bitliktir.Cisc’de komutların boyutu sabit değildir.

3-)Risc’de kodlar basittir.Cisc’de karmaşıktır.

4-) Risc’de program derlenince daha fazla makine kodu olacağından Cisc’ e göre daha fazla alan kapsar.

5-)Cisc’de belleğin şu gözü ile bu gözünü topla şuraya yaz vardır.Risc’de yoktur.

6-)Risc ‘de CPU’daki komut işleme daha hızlı oalcağından bu hızda çalışan CPU’ya hızlı RAM ve büyük önbelleklere ihtiyaç vardır.

**[MAL ‘da 2’ye Tümleme Programı](http://cengaverhero.blogspot.com/2009/06/mal-da-2ye-tumleme-program.html)**

*Gönderen E-Hero*

*Etiketler:* [*Assembly*](http://cengaverhero.blogspot.com/search/label/Assembly) */ Yorumlar: (*[*0*](http://cengaverhero.blogspot.com/2009/06/mal-da-2ye-tumleme-program.html#comments)*)*

.data

number: .word 0xFFFFFF64

mask: .word 0x80000000

temp: .word

.text

la $10,number

lw $11,0($10)

add $12,$0,$0

lw $13,4($10)

and $10,$11,$13

beq $10,$0,finish

loop: and $10,$11,$13

bne $10,$0,git

or $12,$12,$13

git: srl $13,$13,1

bne $13,$0,loop

finish: add $12,$12,1

sw $12,temp

put temp

done

Yukarıdaki kodlar number diye tanımladığımız değişkenin binary karşılığının ilk biti 1 mi önce ona bakar.1 değilse çıktıyı direk 1 verir.Eğer ilk bit 1 ise yani hexadecimal sayı olarak ilk sayı 7 den büyükse sayıdaki 1 leri 0, 0’ları 1 yapar.Sonra 1 ekler çıkan binary sayının decimal karşılığını bize verir.Yani 2’ye tümleme yapar.Yukarıdaki programı çalıştırılınca çıkan sonuç 156 olacaktır.Number değişkenine verilen sayı 15 olarak değiştirilmiş olsaydı sonuç 1 olacaktı.

**[Mips – Risc Mimarisinde Assembly Dili](http://cengaverhero.blogspot.com/2009/06/mips-risc-mimarisinde-assembly-dili.html)**

**Mips Risc mimarisinde Assembly dili 3’e ayrılır.Bunlar SAL,MAL ve TAL’dır.**

**SAL (SİMPLE ABSTRACT LANGUAGE):Gerçek bir assembly dilinin tüm özelliklerini içerir.Ancak gerçek bir Assembly dili değildir.Yüksek seviyeli diller ile makine dili arasınbdaki voşluğu doldurmak için kullanılır.Mal programlamadan daha kolaydır.SAL soyut assembly dilidir.Direkt ikili gösterim karşılığı yoktur.**

**MAL (MORE ABSTRACT LANGUAGE):Mips Risc mimarisinda Mal programı önce TAL’a daha sonrada TAL’daki her komut Mips Risc makine koduna çevrilir.Mal’da programlama Tal dan daha kolaydır.Her ne kadar çoğu MAL komutlarının birebir gerçek S da sahiptir.**

**TAL (TRUE ABSTRACT LANGUAGE):ISA’daki komutlardan oluşur.Tal’dan daha aşağı seviyede programlama yapılamaz.3 operandlı komutlar; Oysa MAL’da 2 veya 3 operandlı komutlar kullanılır.TAL’da MAL’ın aksine sabitler kullanılır.MIPS RISC işlemcisinin gerçek assembly dilidir.SAL veya MAL programı önce TAL programına dönüştürüldükten sonra makine koduna dönüştürülebilir.**