①

```
         .data

0   x:    .word    50

4   y:    .word    1024


         .text


-start    lui    $5, 0x1001
          lw     $3, 0[$5]
          lw     $4, 4[$5]        // l1

          sub    $5, $4, $3

          bltz   $5, etiket       // l2
          add    $5, $3, $0
          add    $3, $4, $0
          add    $4, $5, $0

etiket:   div    $3, $4
          mfhi   $5
          beq    $5, $0, exit

          add    $3, $4, $0
          add    $4, $5, $0
          beq    $0, $0, etiket   // l3

exit:     done
```

→ Data sepmentin
başlangıç adresi
0x1001000 olmak üzere,

Yukarıdaki kod
sonlandığında
$3, $4 repisterlerinin
içeriği ne olur,
16'lik tabanda yazınız.

y = 1024
x = 50

$5	$4	$3
sub y		x → 1. adım
y-x		
x	x	y
¦	¦	1024/50
¦	¦	= 24+mfhi
↓	↓	↓ ⎤ etikete
24	24	50 ⎦ döngü
1.Döngü ↓	↓	↓
2	2	24
2.Döngü ↓	↓	↓
0		
↓		
exit		

$4 = 2 → 2·16°
$3 = 24 → 1·16¹ + 8·16°

→ 0x2
→) 0x18

①

② a) Birinci sorudaki ℓ1 komutunun kodunu hexadecimal olarak yaz.

→ ℓ1 = lw $4, 4($5) ⇒ R_t I (R_b)

$R_t = \$4 = 4 = 0100$
$R_b = \$5 = 5 = 0101$
$I = 4 = 0100$

Kağıtta olan ↓

1000 11bb bbbt tttt iiii iiii iiii iiii

⇒ 1000 1100 1010 0100 0000 0000 0000 0100

8   C   A   4   0   0   0   4   → 0x8CA40004

b) Birinci sorudaki ℓ2 komutunun kodunu hexadecimal olarak yaz.

→ ℓ2 = bltz $5, etiket ⇒ R_s, I

$R_s = \$5 = 5 ⇒ 0101$

$I = 4-1 = 3$ → ℓ2 komutundan etikete kadar sayıp 1 çıkardık

0011

Aşağı inerken çıkar
yukarı giderken topla

Kağıttan ↓ 0000 01ss sss0 0000 iiii iiii iiii iiii

⇒ 0000 0100 1010 0000 0000 0000 0000 0011

0   4   A   0   0   0   0   3   → 0x04A00003

c) ℓ3 komutunun kodunu hexadecimal olarak yaz.

NOT ℓ3 komutu etiketten daha aşağıda olduğundan
bu sefer yukarı doğru gideceğiz.

→ ℓ3 = beq $0, $0, etiket → R_s, R_t, I

0   0   5+1 = ⑥

0000 0000 0110

Kağıttan ↓ 0001 00ss ssst tttt iiii iiii iiii iiii

⇒ 0001 0000 0000 0000 0000 0000 0000 0110

1111 1111 1111 1001
                   1
                   +
⇒ 0001 0000 0000 0000 1111 1111 1111 1010
   1   0   0   0   F   F   F   A   → 0x1000FFFA

d) text segment 0x00400000 adresinden başladığına göre
l3 komutunun adresini hexadecimal yaz.

0x00400000 ,
. . . . 4)
. . . . 8)
. . . . C)
. . . 10)

|

_____

0x00400034

Not: l3'e kadar 13 satır var

$13 \times 4 = 52$
↓
$3.16 + 4.16^{\circ}$
↓
0x00400034

③ Tek duyarlı kayan noktalı sayı temsilini kullanarak topla,
Sonucunu hexadecimale çevir.

56,125 + 100 → 64 + 32 + 4

111000   0,125 × 2 = 0,25   1100100
            0,25 × 2 = 0,5
            0,5 × 2 = 1   → Gerisini
                                     yap
111000,001

↳ 1,11000001 × 2⁵ → $1,11000001 \times 2^5$

④ Tamsayıların bellekte 4 byte yer kapladığı düşünülerek;

a) Aşağıdaki veri tanımlamalarını TAL'da yap

```
int i;
int A[64], int B[64];
```
=)

i .word
A . space 256 (.word 0:64)
B . space 256 (.word 0:64)

Alternatif
çözüm

③

The TAL Instruction Set

Assembly Language Format	Effect
R ← M[[R] + (l₁₆)¹⁶ ‖ l₁₆₀]	
m[[R] + (l₁₆)¹⁶ ‖ l₁₆ ₀h)¹⁴ ‖	

b.) Veri tanımlanalar için text segmentte toplam kaç byte yer ayrılır?

→ 4 byte i; 256 byte A, 256 byte B

⇒ <u>516 byte</u>

c-) A[0] elemanının bellek adresini hexadecimal göster

a) başlangıç   0x10010000 → i

b) A[0] → <u>0x10010004</u>

d.) B[40] elemanının bellek adresini hexadecimal göster

i + A[64] + B[40]

4 + 256 + 160 = 420   →1·256 + 10·16 + 4

1   A   4   → <u>0x100101A4</u>

④