

1 17: 7A 06 00 02 0A 2A
21: 6E 6A 00 07
22: 63 9A
25: 40 F0
28: 00 03 02 05 01 04 FF
29: -

Hypyn pituus on $F0 = -16$ askelta seuraavan käskyn alusta lukien.

EQU ei tuota koodia, se vain määrittelee symbolin.

2 Tarkoitus on tutkia LUVUT-taulukon alemman rivin lukujen sitä bittiä, joka on ilmoitettu vastaavassa kohdassa ylempällä rivillä. Kun kyseinen bitti on ykkönen, BEQ-ehto toteutuu ja luku lisätään summaan. Summaksi saadaan siten $0B + FE = 09$. BEQ-ehdon tulkitseminen toisin päin on myös hyväksytty.

0B	=	0000	1011	(0)
F1	=	1111	0001	(3)
02	=	0000	0010	(2)
06	=	0000	0110	(5)
05	=	0000	0101	(1)
FE	=	1111	1110	(4)

Tehtävään pujahtanut virhe oli siirtänyt rekistereiden R1L ja R2L lataamisen väärään järjestykseen, mistä seuraa osoittimen kasvatus liian aikaisin. Ohjelma siis tutkii oikeasti vasta toisen luvun (F1) nollatta bittiä jne. Tällöin tulokseksi saadaan $F1 + 6 + FE = F5$ (mikäli taulukon jälkeen seuraavan muistipaikan lukua ei oteta mukaan).

3 Laskettavana on summa $1^2 + 2^2 + 3^2 + \dots + n^2$.

```

org      h'20A10

start:   mov.b  #8, R0L      ;testipääohjelmassa n = 8
         bsr    sigma
         bra   $

sigma:   xor.l  ER1, ER1     ;summa nollaksi
         xor.b  R2L, R2L     ;aluksi k = 0

loop:    cmp.b  R2L, R0L     ;k = n ?
         beq   valmis
         inc.b  R2L          ;k = k+1
         mov.b  R2L, R3L
         mulx.b R2L, R3      ;k * k
         extu.l ER3          ;venytetään 32-bittiseksi
         add.l  ER3, ER1     ;lisätään summaan
         bra   loop

valmis:  rts

```

F1	=	1111	0001	(0)
02	=	0000	0010	(3)
06	=	0000	0110	(2)
05	=	0000	0101	(5)
FE	=	1111	1110	(1)
xx	=	xxxx	xxxx	(4)

4 Taulukkoon SEGMENTS on kirjoitettu kunkin numeron koodi, joka ilmoittaa, mitkä ledit palavat (0) ja mitkä ovat pimeitä (1). Esimerkiksi numerossa 0 vain segmentti 'g' on pimeä, joten sen koodiksi tulee 0100 0000 = h'40.

```

org      h'20A10

wait:    btst   #6, @SCI_SSR1      ;odotetaan merkkiä päätteeltä
         beq   wait
         mov.b @SCI_RDR1, R0L      ;luetaan näppäily merkki
         bclr  #6, @SCI_SSR1      ;nollataan vastaanottimen tilatieto
         and.b #h'0F, R0L         ;ASCII-koodi lukuarvoksi
         extu.w R0
         extu.l ERO                ;venytetään 32-bittiseksi
         mov.b @(segments,ERO), R1L ;haetaan taulukosta 7-segmenttikoodi
         mov.b R1L, @P6DR          ;sytytetään ledit koodin mukaisesti
         bra   wait

segments: dc.b  h'40, h'79, h'24, h'30, h'19
         dc.b  h'12, h'02, h'58, h'00, h'10

```

5

