

## Arvo- ja viiteparametri aliohjelmassa

Laaditaan aliohjelma Laske(int n, int &x). Koska Javassa ei ole viiteparametreja, sovitaan siitä, että pseudokielessä viiteparametri ilmaistaan symbolin ’&’ avulla (kuten C++:ssa). Aliohjelma Laske suorittaa yksinkertaisen aritmeettisen laskutoimituksen, jonka tulos palautetaan x:ssä:

$$x = n + 10;$$

Pääohjelma vain kutsuu aliohjelmaa kerran n:n arvolla 15 ja viiteparametrilla M, missä M on pääohjelman muuttuja (siis ”M” vastaa muuttujaa ”x” pari riviä ylempänä). Laskettu tulos myös tulostetaan näytölle.

### Pääohjelma:

	<b>M DS 1</b>	Varataan tilaa muuttujalle M. Huomaa, että M on yhden kokoinen taulukko (data segment), ei siis yksittäinen muuttuja, jonka arvo on 1. Ei sillä oikeasti ole väliä, se voisi olla myös DC tyyppinen. Mutta jos se olisi isompi taulukko, koko taulukko voitaisiin ”antaa” antamalla sen aloitusosoite aliohjelmalle.
<b>Main</b>	<b>LOAD R1, =15</b>	Ladataan rekisteriin R1 arvo 15. Tämä on siis ”n”.
	<b>PUSH SP, R1</b>	Pushataan 15 pinoon. Tämä olisi voitu tehdä myös suoraan PUSH SP, =15.
	<b>PUSH SP, =M</b>	Pushataan M:n osoite pinoon. Tämä on siis viiteparametri: osoite, jonka osoittamaan paikkaan laskennan tulos talletetaan.
	<b>CALL SP, Laske</b>	Kutsutaan aliohjelmaa ”Laske”.
	<b>LOAD R2, M</b>	Ladataan M:n arvo (joka on siis muuttunut aliohjelman aikana) rekisteriin...
	<b>OUT R2, =CRT</b>	...ja tulostetaan näytölle.
	<b>SVC SP, =HALT</b>	Loppu!

**Aliohjelma:**

	<b>n EQU -3</b>	”n”:n paikka pinossa. (ks. kuva alla)
	<b>x EQU -2</b>	”x”:n paikka pinossa. (taas ks. kuva)
	<b>K EQU 10</b>	Luodaan hivin vuoksi vakio K, jolle annetaan arvoksi 10. Voitaisiin ihan yhtä hyvin käyttää ADD R3, =10 vähän myöhemmin.
<b>Laske</b>	<b>LOAD R3, n(FP)</b>	Ladataan muistipaikassa (FP+n) oleva arvo rekisteriin R3. Eli siis käytännössä pinosta 3 muistipaikkaa taaksepäin FP:n osoittamasta muistipaikasta.
	<b>ADD R3, =K</b>	HUOM! K on EQU käskyllä luotu vakio, siksi pitää olla =K, että saadaan K:n ”arvo”, muuten saataisiin mitä muistipaikassa 10 nyt sattuuukaan olemaan.  EQU käskyllä luodut vakiot tarkoittavat suunnilleen ”kun näet tämän kirjaimen/sanan/lyhenteen, kirjoita sen kohdalle tämä luku”. Toisin kuin DC ja DS komennoilla luodut muuttujat, jotka viittaavat osoitteeseen, johon kyseisen muuttujan arvo on talletettu.
	<b>STORE R3, @x(FP)</b>	Tässä on se uusi asia. M (johon viitataan nimellä x aliohjelmassa) on annettu viiteparametrina, joten pääsemme käsiksi M:n oikeaan arvoon kaksinkertaisen muistiviitteen avulla. Ja siten voimme myös muuttaa sitä. ☺
	<b>EXIT SP, =2</b>	Ja poistutaan aliohjelmasta. EXIT-käskyllehän piti antaa parametrina kuinka monta parametria aliohjelman kutsussa oli, että se voi siirtää pino-osoittimen oikeaan paikkaan.

Luentomonistetta ”Tietokoneen toiminta, Auvo Häkkinen, 1998” mukaillen.

Sivuhuomaus: Vakiota  $K = 10$  olisi voinut käsitellä myös seuraavalla tavalla (titotrainer vaatii tätä tapaa joissain tehtävissä, siitä kyllä sanotaan ohjeissa tai push on tehty valmiissa koodissa).

### Kpoint EQU 1

**PUSH SP, =10**

**LOAD R3, n(FP)**

**ADD R3, Kpoint(FP)**

Eli vakion 10 sijaan olisimme luoneet pointterin, joka osoittaa pinossa olevaan vakioon 10, ja sitten käyttäneet tätä pointteria ihan samoin kuin pointteria n. Tätä tapaa käytettäessä pitää kuitenkin muistaa ennen EXIT-käskyä vähentää pino-osoittimen arvoa (tässä tapauksessa ”SUB SP, =1”, koska pinoon laitettiin yksi arvo, jota ei popattu pois), että EXIT-käsky päättyy oikeaan kohtaan ja osaa palauttaa oikean PC:n arvon aliohjelmasta palattaessa.

Jos tämä unohtuisi, ttk-91 yrittäisi käyttää ensimmäistä aliohjelman parametria vanhana FP:nä ja vanhaa FP:tä vanhana PC:nä, mikä todennäköisesti aiheuttaisi invalid memory address tms. virheen. Tai sitten hyppäisi ties minne, jos osoite sattuisi osoittamaan koodiin, eikä data-alueelle.

Ohjelma + aliohjelma	
15 (arvoparametri n)	← FP - 3
M:n osoite (viiteparametri)	← FP - 2
Vanha PC tallennettu tänne	← FP - 1
Vanha FP tallennettu tänne	← SP, FP
<i>10 (jos käytettäisiin Kpointtia, SP siirtyisi myös osoittamaan tähän)</i>	← FP + 1