

Haarukointi – ohjelmointi

ALGORITMIT MATEMA-
TIKASSA, MAA12

Esimerkki a) Osoita, että funktiolla $f(x) = x + \ln x$ on täsmälleen yksi nol-lakohta.

b) Ohjelmoi laskin haarukoimaan se kuuden desimaalin tarkkuudella.

a) Funktio on määritelty, jatkuva ja deri-voituva, kun $x > 0$. Derivaatta

$$f'(x) = 1 + \frac{1}{x} > 0, \quad \forall x > 0.$$

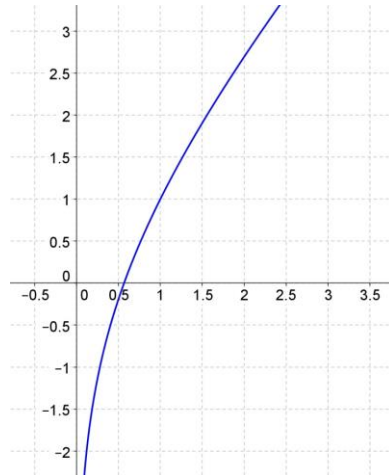
Lisäksi

$$f(0,5) \approx -0,19 < 0$$

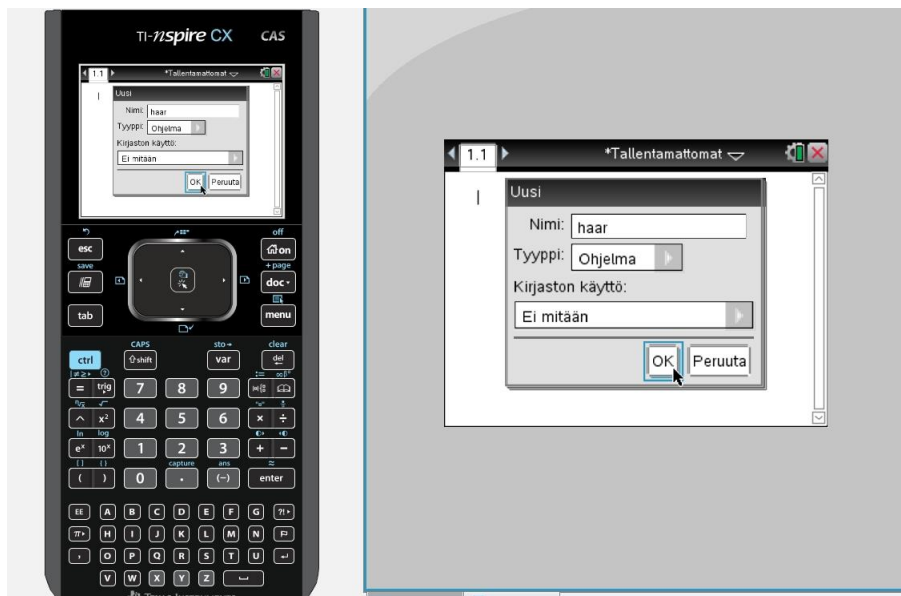
ja

$$f(0,6) \approx 0,09 > 0,$$

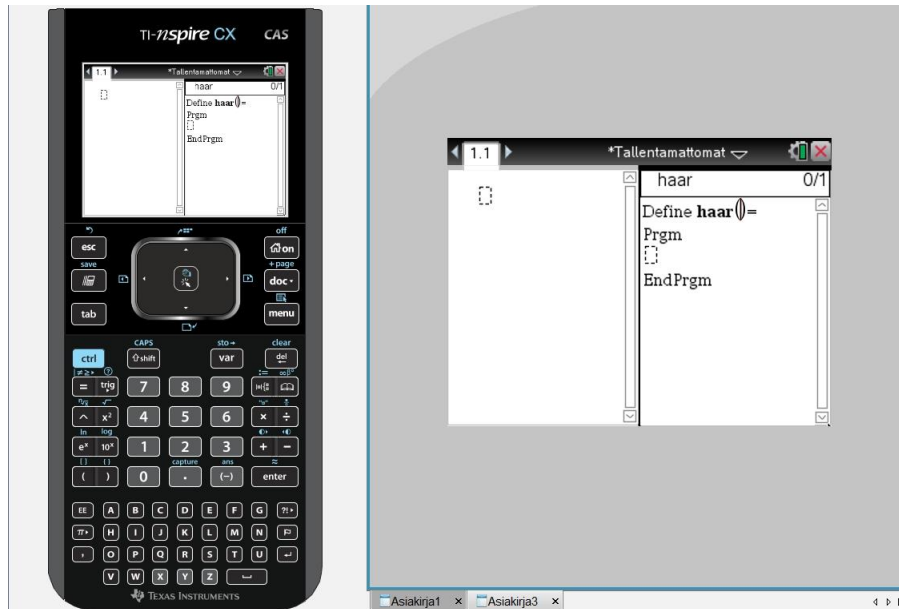
joten Bolzano antaa täsmälleen yhden nollakohdan $\xi \in]0,5; 0,6[$.



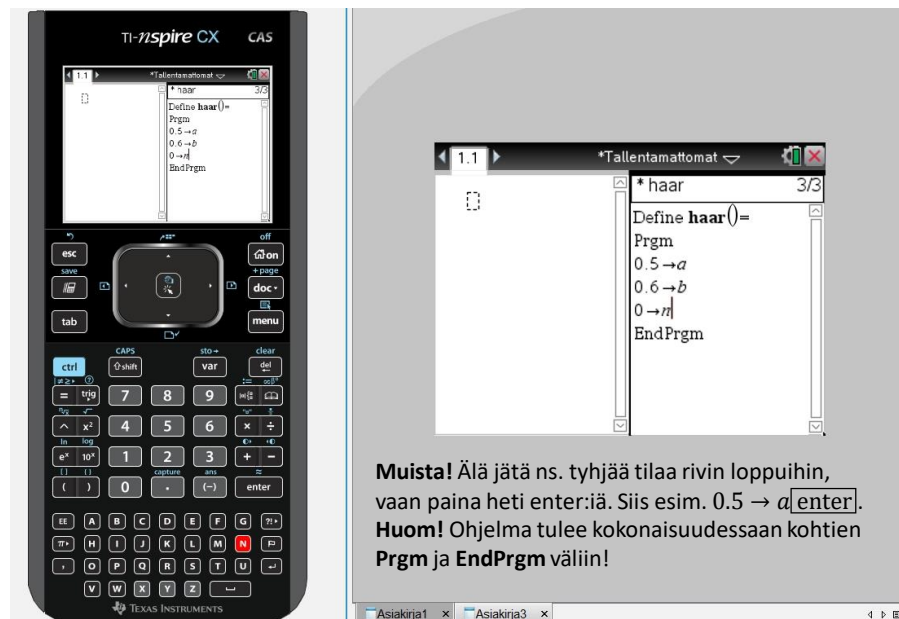
b) Ohjelmointi aloitetaan päävalikosta valitsemalla **Uusi asiakirja**.



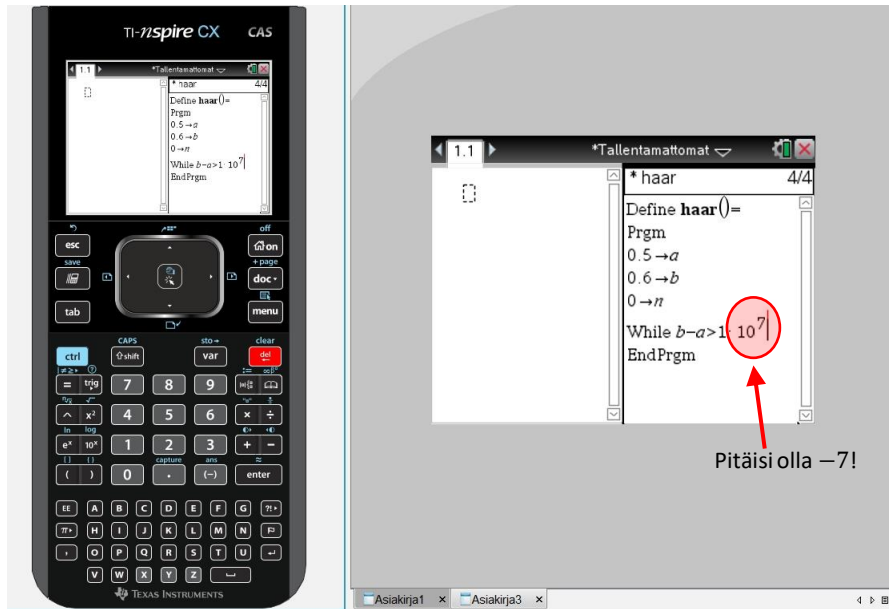
Valitaan **Uusi TI-NSpire™-asiakirja** → **9.Lisää ohjelmaeditori** → **1: Uusi** Annetaan ohjelmalle nimi **haar** tai muu vastaava, joka viittaa haarukointiin (puolitusmenetelmä).



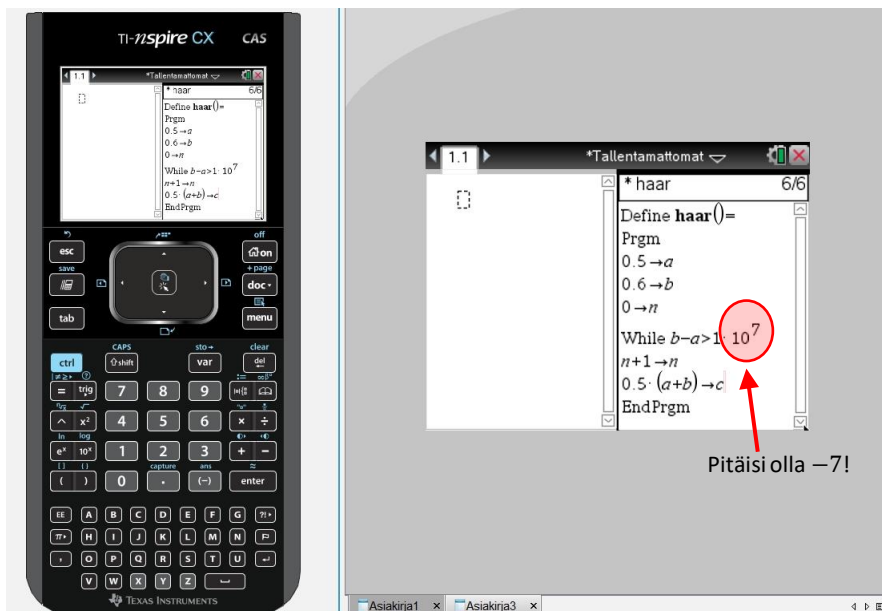
Ohjelman aloitusnäkyä.



Ohjelmakoodin aluksi kerrotaan ala- ja yläkiarvot sekä tallennetaan ne muistipaikkoihin a ja b . Sitten 3. rivillä perustetaan kierroslaskuri, joka nollataan. (Tämä ei ole tehtävän kannalta välttämätöntä, mutta on kiinnostavaa tietää kierrosten lkm.)



Tämän jälkeen luodaan **While**-silmukka, joka pitää päättää **End** -komentoon (myöhemmin), muuten tulee syntaksivirhe. Silmukka suoritetaan, niin kauan kunnes etsityn nollakohdan ylä- ja alalikiarvojen erotus saavuttaa tai alittaa arvon 10^{-7} .



1. Viidennellä rivillä kasvatetaan laskuria yhdellä, eli $n + 1 \rightarrow n$.
2. 6.nella rivillä lasketaan ylä- ja alalikiarvojen keskiarvo ja tallennetaan se muistipaikkaan c , eli $(a + b)/2 \rightarrow c$.

Pitäisi olla $-7!$

```
*Tallentamattomat 11/11
* haar
0 → n
While b-a > 1 107
n+1 → n
0.5 (a+b) → c
If c+ln(c) < 0 Then
c → a
Else
c → b
EndIf
```

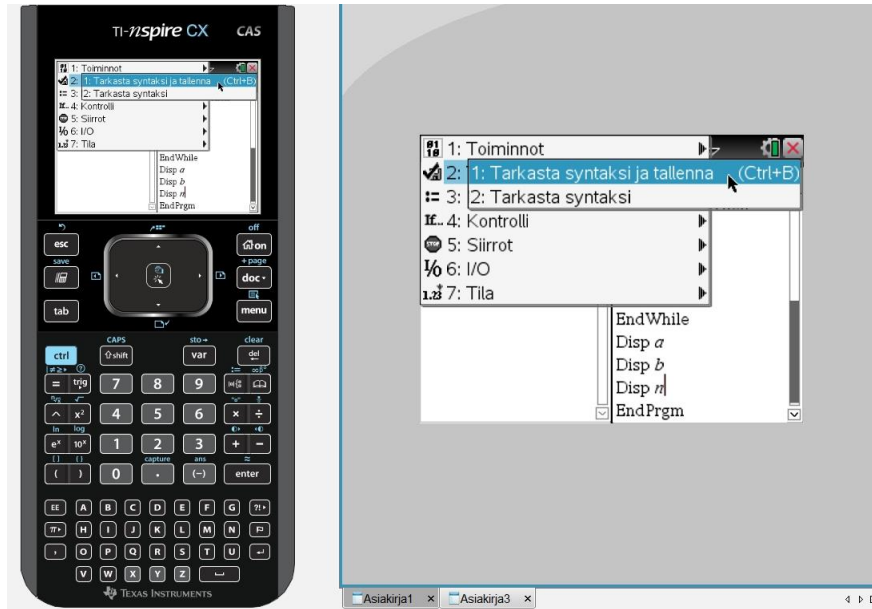
Haarukointitehtävissä (puolitustehtävissä) muutetaan lauseke **If** ja **Then** sanojen välissä, muuttujana c . Ja alkulikiarvot myös (tietysti)!

Välilyönti **If**-sanan jälkeen ja ennen **Then** -sanaa!

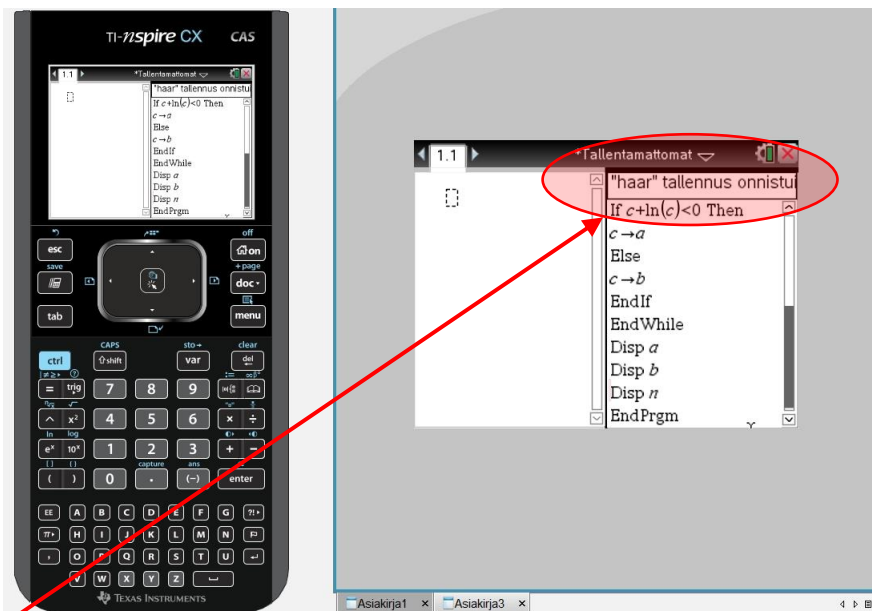
7.nnellä rivillä alkaa algoritmin kolmosvaihetta vastaava **If – Then – Else**-lause, joka päättyy 11.rivin **EndIf**-komenttoon. Funktio f voitaisiin ennen ohjelman kirjoittamista määritellä $y1 = x + \ln x$, jolloin **If** -lauseen eka komento olisi $y1(c) < 0$.

```
*Tallentamattomat 15/15
* haar
If c+ln(c) < 0 Then
c → a
Else
c → b
EndIf
EndWhile
Disp a
Disp b
Disp n
EndPrgm
```

Lopuksi suljetaan **If** -silmukka ja **While**-silmukka sekä näytetään (**Disp**=display) luvut a , b ja n .
Huomaa välilyönti **Disp** -komennon jälkeen!



Kun ohjelma on saatu koodattua, sen syntaksi pitää tarkistaa ja tallentaa. Siis Paina hiiren oikeata ja valitse 5: Tarkasta syntaksi ja tallenna (TAI Menu → 2:Tarkasta syntaksi ja tallenna → 1:Tarkasta syntaksi ja tallenna.)



Ylös tulee ilmoitus: "haar" tallennus onnistui... ja vasemmalla ylhäällä ollut asterix-merkki * häviää.

The image shows a TI-84 Plus CE calculator on the left and a computer screen on the right. The calculator screen displays the following code for the Haar algorithm:

```

"haar" tallennus onnistu
Define haar()=
Prgm
0.5→a
0.6→b
0→n
While b-a>1·10-7
n+1→n
0.5·(a+b)→c
If c+ln(c)<0 Then

```

The computer screen shows the same code being entered into a program editor. The top window shows the code being typed, and the bottom window shows the code after execution, with the output:

```

haar()
0.567143
0.567143
20
Valmis

```

The text below the calculator and screen reads: "Viedään kursori vasemman puoleisen sarakkeen kohdalle ja kirjoitetaan ohjelman nimi+sulut, eli haar(). Sitten vaan painetaan enteriä → vastaus."