

Tehtävien ratkaisut tulee olla tehtynä Googlen Docs-ohjelmalla. Liitä vastauksiisi kuvia TI-*n*spire ohjelmalla tai IDLE ohjelmalla tuotetuista koodeista.

→ Palauta vastaukset PEDAn palautuskansioon annettuun päivämäärään mennessä!

1. Mitä virheitä seuraavissa koodeissa on, kerro rivi ja kuinka korjaisit koodin oikeaksi eli toimivaksi:

a) Kaksi asiaa väärin

```
luku=float(input("Anna positiivinen kokonaisluku."))
print("Luvun",luku,"kertotaulu:")
for i in range(1,11,1):
    ♦♦print(i,"x",luku,"=","i*luku)
#-----
```

b) Kaksi asiaa väärin

```
import math
luku =int(input("Anna positiivinen kokonaisluku >1."))
while luku <=1:
    ♦♦luku=int(input("Anna positiivinen kokonaisluku >1."))
print(luku,end=" = ")
i=2
while i <= luku
    ♦♦if luku/i ==1:
        ♦♦♦♦print(i)
        ♦♦♦♦break
    ♦♦if luku%i==0:
        ♦♦♦♦print(i,end=" x ")
        ♦♦♦♦luku=luku/i
        ♦♦♦♦i=i
    ♦♦else:
        ♦♦♦♦i=i+1
```

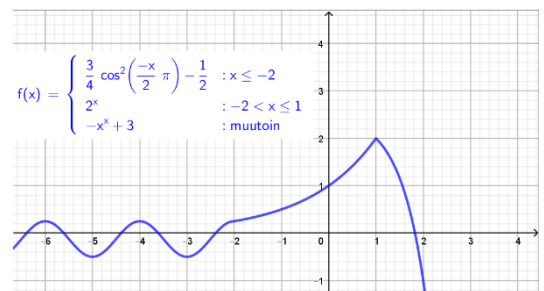
2. Funktiot ovat ohjelmoinnissa matematiikan funktioita laajempi käsite. Ne voivat suorittaa komentoja tai antaa paluuarvon `return`-komennolla matemaattisten funktioiden tapaan. Pythonissa funktiot määritellään `def` -komennolla. Funktioiden suorittaminen on mahdollista myös Python Shellissä.

→ Käytä TI:ssä Python-tiedoston luonnissa “Tietojen jakaminen” -tyyppiä.

**Muistakaa, jos koodin ajaminen johtaa ikuisen silmukkaan, niin F12 keskeyttää koodin ajon!**

a) Luo paloittain määritelty funktio

$$f: f(x) = \begin{cases} \frac{3}{4} \cos^2\left(\frac{-x}{2}\pi\right) - \frac{1}{2}, & x \leq -2 \\ 2^x, & -2 < x \leq 1 \\ -x^x + 3, & 1 < x \end{cases}$$



ja etsi funktion  $f$  jokin nollakohta Newtonin menetelmän

avulla Pythonkoodia käyttäen. Newtonin menetelmä on seuraava:

- Luodaan alkulikiarvo, esim.  $x = 1,5$

- Lasketaan kaavalla  $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$  seuraavat likiarvot, missä indeksi  $n - 1$  kuvaa edellistä likiarvoa.

- Lisätietoa esimerkiksi kirjan JUURI 11 sivulta 138 ja 141.

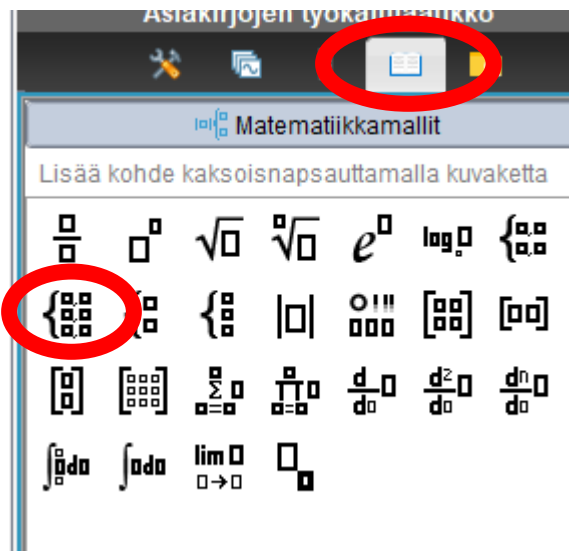
**Ohjeita:** Luokaa TI:ssä muistiinpanot-sovelluksessa tai laskin-sovelluksessa funktio  $f$  ja sen derivaatta. Muista

$f(x) := 2 \cdot x - 1$  Valmis  
 $:=$  komento. Siis esim.

Paloittain määritelty funktio TI:ssä → Käyttäkää valmista Matematiikkamalleista löytyvää komentoa. Tulis näyttää tältä

$$f(x) := \begin{cases} \frac{3}{4} \cdot \left( \cos\left(\frac{-x}{2} \cdot \pi\right) \right)^2 - \frac{1}{2}, & x \leq -2 \\ 2^x, & -2 < x \leq 1 \\ -x^x + 3, & 1 < x \end{cases} \quad \text{Valmis}$$

$$g(x) := \frac{d}{dx}(f(x)) \quad \text{Valmis}$$



**Huomio!**

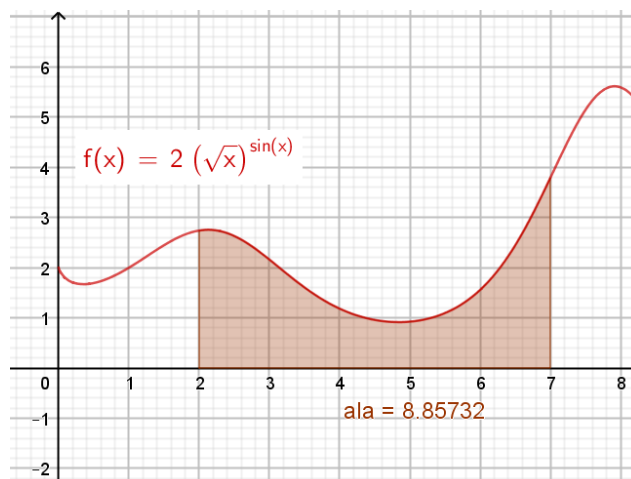
Jos välillä tyhjennätte muistipaikat, mikäli tullut virhe funktion  $f$  lausekkeisiin, niin ottakaa uusi TI-tiedosto. Koodi ilmoittaa virhettä, ilmeisesti ongelma on siinä Data sharing:ssä eli tietojen jakamisessa.

Yrittäkää luoda silmukka, jossa muistipaikkaan tallennetulla arvolla suoritetaan lasku ja se tallentuu uudeksi arvoksi kyseiseen muistipaikkaan. Keskeyttäkää silmukka, kun kahden peräkkäisen niin kutsutun iteraatioarvon eli muistipaikkaan tallennettujen arvojen välinen ero (etäisyys...ainiinjoo tarvitaan...) on vaikka pienempi kuin  $10^{-6}$ . Palauttakaa funktion nollakohdan arvoksi viimeinen muistipaikkaan tallennettu arvo. Voitte lopuksi verrata sitä ns. tarkkaan arvoon. Halutessanne voitte myös ilmaista, montako iteraatiokierrosta meni haluttuun tarkkuuteen pääsemiseksi (tämä on siis vapaaehtoista).

GEOlla saa piirtää kuvan ja esim. tutkia miksi jotkut alkuarvot toimivat ja toiset eivät. Eli funktion kuvaajalle lisäätkää piste komennolla "Piste objektilla" ja sitten tähän pisteeseen laittakaa tangenti.

**b) Määrätyn integraalin sovellus.**

Määrättyä integraalia voidaan arvioida ns. Simpsonin säännöllä, jossa funktion kuvaajaa arvioidaan paraabelin kaarilla. Teoria hieman sivuutetaan nyt ja otetaan käyttöön tulos (alla).



Tarkastellaan funktiota

$$f: f(x) = 2(\sqrt{x})^{\sin(x)}$$

ja sen määrättyä integraalia

$$\int_2^7 f(x)dx = 8,8573 \dots$$

Simpsonin säännön nojalla pätee

$$\int_2^7 f(x)dx \approx \frac{h}{3}(f(x_0) + 4 \cdot f(x_1) + 2 \cdot f(x_2) + 4 \cdot f(x_3) + \dots + 2 \cdot f(x_{n-2}) + 4 \cdot f(x_{n-1}) + f(x_n)),$$

missä muuttujan  $x$  arvot eli kohdat  $x_n$  ovat:  $x_0 = 2$  välin alkukohta,  $x_1 = x_0 + h$ ,  $x_2 = x_0 + 2h$ , jne. ja

$x_n = 7 = 2 + n \cdot h$  välin päätekohta ja  $n$  kuvaa siis välin  $[2,7]$  osavälien lukumäärää (tasavälinen jako)

ja  $h = \frac{7-2}{n}$  on yhden osavälin pituus.

Huomaa, että päätekohtissa  $x = 2$  ja  $x = 7$  funktion arvo lasketaan vain kerran, mutta muissa kohdissa joko 4 tai 2 kertaa.

**Ohjeita.** Luokaa ensin TI:ssä muistiinpanot-sovelluksessa tai laskin-sovelluksessa funktio  $f$  muistiin.

Luokaa ohjelma, joka kysyy käyttäjältä integroinnin rajat, osavälien lukumäärän (**nyt siis oltava parillinen**

**positiivinen kokonaisluku**. Ohjelman tulee tarkistaa: yläraja  $>$  alaraja, annetun osavälien määrää

kuvaavan luvun parillisuus ja positiivisuus...esim. onko jakojäännös 1, kun jaetaan kakkosella tms.)

Tallentakaa osavälin  $h$  arvo muistipaikkaan ja sulkujen sisällä olevan lausekkeen

$$f(x_0) + 4 \cdot f(x_1) + 2 \cdot f(x_2) + 4 \cdot f(x_3) + \dots + 2 \cdot f(x_{n-2}) + 4 \cdot f(x_{n-1}) + f(x_n)$$

arvon saatte, kun muodostatte kolme eri arvoa, missä yhden saatte päätekohtista ja kaksi muuta kun

muodostatte kaksi silmukkaa ja laskette lopuksi tulokset yhteen. Muistakaa ns. nollata laskurit ensin sil-

mukoiden tekoa. Ilmoita vastaus 3 desimaaliin pyöristettynä.

Laskekaa vielä TI:n laskinsovelluksen antama ”tarkka” määrätyn integraalin arvo.