

1. Harjoitellaan funktioiden kirjoittamista. Tee ohjelma, jossa määrittelet funktiot, jotka
  - (a) tulostaa ruudulle viestin.
  - (b) lukee käyttäjältä kokonaisluvun ja palauttaa sen.
  - (c) saa argumentiksi reaalityluvun  $x$  ja palauttaa  $1/(1-x)$  (H4T4)
  - (d) saa argumenteikseen 2 reaalitylukua ja palauttaa niistä suuremmanKirjoita pääohjelma, jossa testaat kirjoittamiasi funktiota.
2. (H4T3) Tee ohjelma, joka pyytää reaalityluvun  $x$  ja kokonaisluvun  $n$ , ja laskee summan

$$\sum_{k=0}^n \frac{x^k}{k!}$$

3. Tee tehtävän 2 summasta funktio, nimeltään esim. `exps`, joka ottaa argumenteikseen  $x:n$  ja  $n:n$  ja palauttaa summan arvon. Vertaa funktiosi antamia tuloksia `exp(x)` funktion antamiin, tulosta ruudulle funktiosi tulos, `exp(x)`, ja funktiosi ja `exp(x):n` erotus. (Funktio `exp(x)` saadaan lisäämällä `#include <math.h>` rivi ohjelman alkuun ja se siis laskee  $e^x$  :n arvon)
4. Tee ohjelma, joka laskee pankkitilille talletetun summan kasvun vuosittain omassa funktiossaan. Käyttäjä antaa pääfunktiossa tilille alkupääoman, korkoprosentin ja talletusajan vuosina. Summa kasvaa joka vuosi seuraavan kaavan mukaisesti:

$$\text{pääoma vuoden lopussa} = \text{pääoma vuoden alussa} \cdot \frac{100 + \text{korkoprosentti}}{100}$$

Funktio voisi olla muotoa `double korkoaKorolle(double alkupaaoma, double korkoprosentti, int talletusaika);` Funktio palauttaa talletusajan lopussa tilillä oleven summan. Tulosta tämä summa pääohjelmassa.

Lisätehtäviä:

5. Lisätehtävä: Muuta tehtävän 2 summan laskemista siten, että summaa lasketaan kunnes tulos ei enää muutu.
6. Häijy lisätehtävä: Funktiot voivat myös kutsua itseään. Tällaisia funktioita sanotaan rekursiivisiksi, ja joissain ohjelmointikielissä tavallinen tapa ohjelmoida toisto on käyttää rekursiivistä funktiota. Laske kertoman arvo käyttämällä rekursiivistä funktiota. Apuna voit käyttää seuraavaa kertoman matemaattista esitystä (merkitään  $n! = f(n)$ ):

$$f(n) = \begin{cases} 1, & n = 0 \\ n f(n-1), & n > 0 \end{cases}$$