

1. Kirjoita funktio, nimeltään esim. `expn`, joka ottaa argumenteikseen x :n ja n :n ja palauttaa summan

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

arvon.

Harjoituksissa esitetään ratkaisu käyttämällä `for`-silmukkaa.

Tehtäväksesi jää kirjoittaa uusi funktio, esim `expw(x)`, joka laskee saman summan, mutta käyttää `while`-silmukkaa. 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)

2. Tee ohjelma, joka laskee pankkitilille talletetun summan kasvun `for`-silmukassa 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 laskepääoma(double alkupääoma, double korkoprosentti, int talletusaika)`; Funktio palauttaa talletusajan lopussa tilillä oleven summan. Tulosta tämä summa pääohjelmassa.

3. Kirjoita ohjelma, joka tulostaa seuraavien neljän lausekkeen arvot kolmeen sarakkeeseen siten, että x saa arvot $-1.0, -0.9, \dots, 2.9, 3$:
 $\sin x$, $x^{-2.5}$, $\lceil x \rceil$ (lähinnä suurempi kok. luku) ja $\log x$. (Vihje: `for`-silmukka!). Mieti, miten voisit soveltaa `top-down design` menetelmää tässä tehtävässä.