

1. Kirjoita ohjelma, joka laskee kertoman $n! = 1 \cdot 2 \cdot \dots \cdot (n-1) \cdot n$ käyttäen while, do-while ja for silmukoita, ja jotain käyttäjän antamaa lukua n .
2. Kirjoita ohjelma, joka lukee käyttäjältä lukuja, ja tulostaa viittä suurenpien lukujen lukumäärän. Ohjelma pysähtyy, kun annetaan negatiivinen luku.
3. Tee ohjelma, joka pyytää reaaliluvun x ja kokonaisluvun n , ja laskee summan

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

4. Kirjoita ohjelma, johon kirjoitat funktion joka laskee lausekkeen $1/(1-x)$ arvon. Funktio saa x :n argumenttinaan ja antaa tuloksen paluuarvonaan.
5. 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)
6. Lisätehtävä: Muuta tehtävän 5 summan laskemista siten, että summaa lasketaan kunnes tulos ei enää muutu.
7. 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 $n! = f(n)$):

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