Tankebry: Rekursjon

Læringsmål:

- Rekursjon
- Betingelser
- Filbehandling

Pensum:

- 3.7 - User-Defined Functions That Return a Single Value
- 4.1 - The if statement
- 4.2 - The if-else statement
- 4.3 - Nested if-else statements
- 5.1 - The for loop
- 9.1 - Lower-Level File I/O Functions
- 10.5 - Recursive Functions

Rekursive funksjoner er som Inception: Det tar tid før man forstår hva som skjer. Det kan være lurt a skisser hva som skjer ved en forenkling av hver problemstilling med en rekursjonsdybde på 4. Begynn gjerne ved det innerste funksjonskallet og jobb deg utover.

a)

Fibonaccitallene er definert som følger:

$$ f(n) = \begin{cases} f_{n-1} + f_{n-2} & n > 2\\ 1 & n = 2\\ 1 & n = 1 \end{cases} $$

For eksempel er $f_3 = f_1 + f_2 = 1 + 1 = 2$. Dermed blir begynnelsen av rekken slik: 1, 1, 2, 3, 5, 8, 13, 21... Lag den rekursive funksjonen `fibonacci` som tar tallet **n** som parameter og returnerer det n-te elementet i fibonacci-følgen.

b)

Skriv om funksjonen i **a)** slik at den returnerer en liste over de **n** første fibonaccitallene. Lag funksjonen fibStore, som bruker funksjonen forklart tidligere, og lagrer fibonacci-følgen i en fil med ett tall per linje.

c)

Lag funksjonen `fact` som tar tallet n som parameter og returnerer resultatet av den matematiske operasjonen n!. Funksjonen er definert slik:

$$ f(n) = \begin{cases} 1 & n <= 1 \\ n(n-1)! & \text{ellers}\\ \end{cases} $$

d)

Lag funksjonen des2bin(descimal) som tar inn et positivt heltall (eller 0) og returnerer den binære representasjonen av tallet som en tekststreng. Funksjonen skal være rekursiv, og for hvert rekursive kall skal funksjonen finne det binære sifferet lengst til høyre i resultatet. Funksjonen kan implementeres etter følgende rekursjonsskjema:

$$ f(n) = \begin{cases} '0' & n = 0 \\ '1' & n = 1 \\ \text{[des2bin(}\frac{n}{2}\text{), '0']} & n:\text{ partall} \\ \text{[des2bin(}\frac{n-1}{2}\text{), '1']} & n:\text{ oddetall}\\ \end{cases} $$

Test funksjonen slik:

```
des2bin (0)         % '0'
des2bin (1)         % '1'
des2bin (2)         % '10'
des2bin (127)         % '1111111 '
```

e)

Lag funksjonen `towerOfHanoi(n, source, dest, temp)`. Funksjonen skal skrive ut løsningen på Tower of Hanoi problemet. Se wikipedia for denisjon av problemet. Eksempelutskrift for `towerOfHanoi(3, 1, 3, 2)` er:

```
Flytt fra 1 til 3
Flytt fra 1 til 2
Flytt fra 3 til 2
Flytt fra 1 til 3
Flytt fra 2 til 1
Flytt fra 2 til 3
Flytt fra 1 til 3
```

f)

Lag funksjonen `recSine(x)`. Denne skal regne ut sin(x) rekursivt ved at: $\sin(x) = 3\sin(x/3)-4(\sin(x/3))^3$

Benytt også at: $sin(x) \approx x \text{ når } \lvert x \rvert << 1$. Bruk den innebygde funksjonen sin(x) til å bekrefte svaret.

g)

Lag en funksjon som rekursivt multipliserer rekken: $(1+1/1^2)(1+1/2^2)(1+1/3^2) ...$

Avslutt iterasjonen når $(1+1/N^2) < 1 + \text{tol}$, hvor `tol` er feiltoleransen

Skriv også ut resultatet for hver rekursjon.