

# Rekursion

*I denna laboration kommer vi att titta lite närmare på hur rekursion fungerar.*

## Mål

Du ska efter denna laboration kunna:

- förstå hur rekursion går till
- använda rekursion
- förstå när man skall undvika rekursion
- veta vad begreppet ändrekursion betyder
- använda iteration i kombination med rekursion

Tonvikt läggs på:

- struktureringen av problemet
- läsbarhet av programkod

## Uppgift 1 (frivillig)

Du skall skriva den rekursiva implementationen av funktionen som har deklaration enligt följande:

```
function Factorial(N : in Natural) return Positive;
```

Funktionen skall beräkna  $N!$  utifrån det  $N$  som kommer in som parameter. För att visa att funktionen fungerar krävs förstås ett huvudprogram som anropar funktionen.

## Uppgift 2 (frivillig)

Du skall skriva de rekursiva implementationen av funktionen som har deklaration enligt följande:

```
function Power(X : in Float;  
              N : in Integer) return Float;
```

Funktionen skall beräkna  $X^N$ .

## Uppgift 3

Du skall skriva en funktion som beräknar det  $N$ :te talet i Fibonacci-serien. Indata till funktionen skall vara  $N$  och funktionen skall givetvis vara rekursiv. Funktionen skall heta `Fib`.

Definition av Fibonacci-serien:

```
Fib(1) = 1  
Fib(2) = 1  
Fib(N) = Fib(N - 1) + Fib(N - 2)
```

Serien börjar alltså som följer:

```
1 1 2 3 5 8 13 21 34 ...
```

Testa med ett stort tal på  $N$  och se vad som händer. Varför blir det så?

## Uppgift 4

Skriv ett program som läser in ett antal rader med tre tecken långa strängar. När inmatningen "---" kommer skall alla tidigare inmatningar skrivas ut i omvänd ordning. Det kan vara godtyckligt många rader att mata in (vilket gör att du inte kan lagra dem i ett fördimensionerat fält).

Exempel på programkörning:

Mata in ett antal rader (3 tecken per rad, avsluta med "---"):

**aaa**

**bbb**

**ccc**

**---**

Du matade in följande rader:

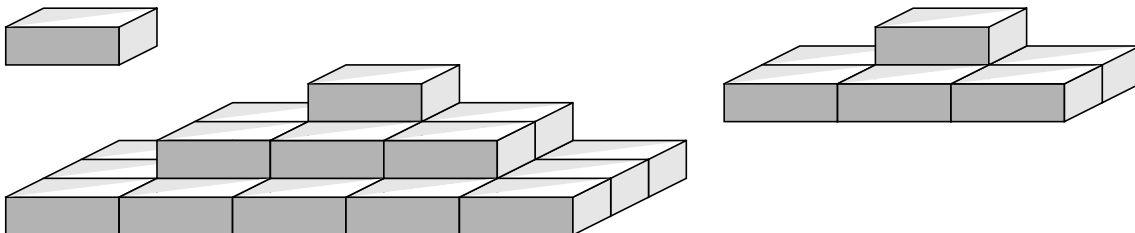
ccc

bbb

aaa

## Uppgift 5

När man bygger sitt hus och kommer till den punkt där man skall bygga taket får man lite problem. Man behöver någon form av ställning. Antag att man bygger den av LECA-stenar på så sätt att man kan se det som en trapp oavsett från vilket håll man kommer (lite grann som en kapad pyramid). Se i figuren för att se hur trappen skulle se ut för hushöjderna 2, 3 och 4 (självlklart behövs ingen trappa om man bara bygger ett lager stenar i huset).



Din uppgift är att skriva det program som räknar ut hur många stenar det behövs för att bygga "trappan" givet en viss höjd på huset.

Krav: Du skall ha en rekursiv funktion som beräknar antalet stenar i ditt program.

Körexempel 1:

Mata in husets höjd (HH = antal stenar i höjddled): 10

Det behövs 525 stenar för att bygga trappan.

Körexempel 2:

Mata in husets höjd (HH = antal stenar i höjddled): 100

Det behövs 651750 stenar för att bygga trappan.