Kodeskabeloner til vektorer

Til "Imperative programming" på codelabby.com

Lad os antage, at variablen a enten indeholder en array reference eller en tekststreng.

Næsten alle program-stumper, der gør noget med sådan en vektor, har én af to former:

- Enten løber vi a igennem fra venstre mod højre, mens vi gør noget med `a[i]` undervejs
- Eller også løber vi a igennem fra højre mod venstre, mens vi gør noget med `a[i]` undervejs

Der er ingen grund til at genopfinde den dybe tallerken på en ny måde hver gang, vi gør det her. I stedet vil vi lære at genkende og skrive følgende to kodeskabeloner, som er hinandens spejlbilleder:

Venstre-mod-højre gennemløb (forlæns)

```
let i = 0
  while i < len a
    // gør noget med a[i]
    set i = i + 1
```

Højre-mod-venstre gennemløb (baglæns)

```
let i = len a
  while i > 0
    set i = i - 1
    // gør noget med a[i]
```

I begge tilfælde skal du erstatte kommentaren `// gør noget med a[i]` med noget kode, der afhænger af den konkrete ønskede opførsel af programstumpen. Her skal du tænke selv.

Variant: variabel til længden af vektoren

I venstre-mod-højre gennemløbet beregner vi længden af a i hver iteration. Det giver samme værdi hver gang, så vi kan spare lidt CPU-kræfter ved at indføre en variabel n, der indeholder den værdi:

```
let n = 0
let i = 0
  while i < n
    // gør noget med a[i]
    set i = i + 1
```

Variablen n er ofte nyttig på anden vis også. Nogle gange skal man afbryde programmet med en fejl inden løkken, hvis n er 0. Andre gange skal man sammenligne `i` med n efter løkken for at finde ud af, om koden `// gør noget med a[i]` har afviklet en **`break`**-kommando og dermed afsluttet løkken tidligt.

Efter løkken er `i < n` hvis løkken er blevet afbrudt med **`break`**. Ellers er `i = n`.