

Learning to Program with F#  
Exercises  
Department of Computer Science  
University of Copenhagen

Jon Sparring, Martin Elsmann, Torben Mogensen, Christina Lioma

August 24, 2020

## 0.1 Continued Fractions

### 0.1.1: Skriv en rekursiv funktion

```
cfrac2float : lst:int list -> float
```

som tager en liste af heltal som kædebrøk og udregner det tilsvarende reelle tal.

### 0.1.2: Skriv en rekursiv funktion

```
float2cfrac : x:float -> int list
```

som tager et reelt tal og udregner dens repræsentation som kædebrøk.

### 0.1.3: Skriv en rekursiv funktion

```
frac2cfrac : t:int -> n:int -> int list
```

som tager tæller og nævner i brøken  $t/n$  og udregner dens repræsentation som kædebrøk udelukkende ved brug af heltalstyper.

### 0.1.4: Skriv en rekursiv funktion

```
cfrac2frac : lst:int list -> i:int -> int * int
```

som tager en kædebrøk og et index og returnerer  $t_i/n_i$  approximationen som tuplen  $(t_i, n_i)$ .

### 0.1.5: Saml alle ovenstående funktioner i et bibliotek bestående af dets interface og implementationsfil (`continuedFraction.fsi` `continuedFraction.fs`), og lav en applikationsfil, der udfører en white- og black-box test af funktionerne.