



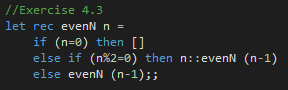
Se slides.



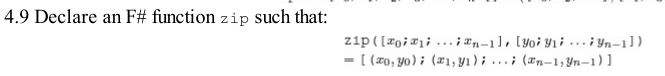


Denne funktion tæller ned, parameteren -1 sørger for dette.

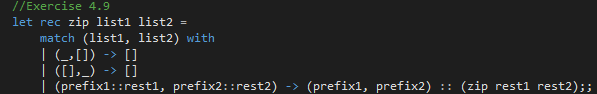


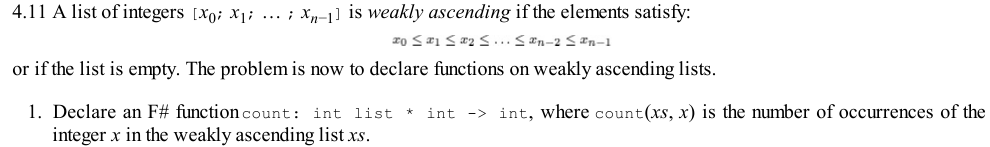


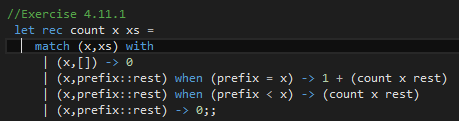
Denne funktion tager et heltal n, og tjekker i hvert trin om n er et lige heltal. Hvis det er lige, så appendes en output liste med n. Dette fortsætter indtil n er lig 0, hvor den tomme liste i stedet, som er base case.



Denne funktion fletter 2 lige lange lister sammen, hvor hvert element-par sættes sammen som et tal-par (tuple).

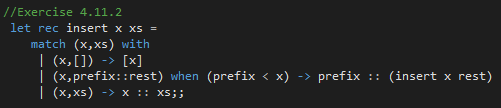




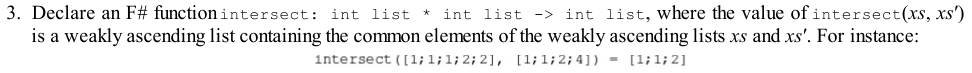
Vi antager at alle lister der arbejdes med er gyldige. 

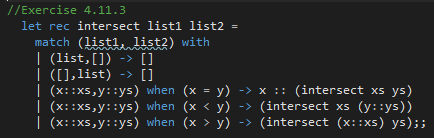
Count er af typen (int,list) → int. Outputtet er antallet af gange x optræder i XS. Hvert element i XS gennemgås, ved at fjerne det forreste element hver rekursion, og dette element sammenlignes med x. Hvis det er lig x, så lægges 1 til resultatet. Hvis ikke, lægges 0 til. Når alle elementerne er gennemgået, vil resten være tom, og 0 lægges til resultatet, og vi er færdige.





Denne tager en liste og et heltal x. Hvis listen er tom, returneres x.  
Taktikken er at indsætte x så tidligt som muligt, så hvis vi møder et element der er lig med, eller større end x, så indsætter vi x før dette element. Til sidst returneres den originale liste, hvor x nu er indsat på den korrekte plads.





Intersect tager de 2 lister, og sammenligner hvert element.  
Hvis de er éns, så lægges det til (x er det samme som y, men her appendes med x).  
Hvis det ene element er mindre end det andet, så smides det væk, og funktionen kaldes rekursivt med de samme liste, dog hvor den ene liste med det mindste element er fjernet.