

# Assignment1 - Addressed Feedback

Victor Vangkilde Jørgensen  
kft410

October 12, 2024

## 1 Omskrevne spørgsmål

1a, 1c, 1e, 2b, 2c, 2d, 3a, 3a

## 2 Gennemgang af feedback til hvert adresseret spørgsmål

### 2.1 Overall impression

#### 2.1.1 Nogle af dine forklaringer bliver lidt lange, hvilket er unødvendigt. Prøv om du kan skrive mere præcist.

Jeg har i min resubmission forkortet en del af de længere forklaringer i håb om, at kommunikere mine hovedpointer nemmere til modtageren.

#### 2.1.2 Husk at ændre templatens til at passe til den specifikke rapport, dvs. afsnit og nummerering.

Hvert afsnit er nu nummereret som opgavens format.

#### 2.1.3 Husk at vi ikke vil mixe actions og computations. Dvs. ingen print statements i vores funktioner.

Dette var det eneste problem, som jeg ikke endnu har kunne løse for rekursive funktioner. Jeg gennemgår de steder jeg har arbejdet med dette problem senere.

#### 2.1.4 Vi vil ikke have nogen hardcoded dele i vores funktioner. I stedet skal funktionerne tage disse dele som argumenter. Vi gør dette for at øge funktionernes genanvendelighed.

Nedeunder adresserer jeg de tilfælde, hvor jeg har hardcoded dele af funktionerne. I den genafleverede src folder og rapport kan man finde de nye, rettede funktioner, der ikke har hardcoded dele.

### 2.2 Feedback on question 1

#### 2.2.1 En tuple bestående af hvilke typer? (1a)

Jeg har i den rettede genaflevering specificeret, hvilke types jeg benytter i mine vektor tuples.

#### 2.2.2 Når man skriver en specification af en funktion skal man skrive hvad en funktion gør, og hvordan den gør det. Dvs. du skal fx inkludere nogle formler osv.

Jeg mindes, at Philippe sagde tallige gange: "A specification (for now) is a short textual description of \*what\* a function does (not how it does it)?" Hvilket også står i vores announcement: "FAQ Assignment 1." Jeg havde derfor ikke skrevet hvordan funktionen specifikt behandler dens input til at

danne et output.

På baggrund af dette, har jeg med al respekt valgt, at ignorere feedback'en vedrørende dette (for nu).

### **2.2.3 Når du forklarer, hvordan du har konstrueret funktionerne, burde du nævne din brug af let bindings.**

Jeg har nu vedlagt en forklaring omkring let bindings tidligere i opgaven end før. Denne forklaring er meget kort, og skal virke som en "teaser" til den længere forklaring senere i opgaven.

## **2.3 Feedback on question 2**

### **2.3.1 (Count function:) I stedet for at hardcode det spørgsmål og svar vi ønsker i selve funktionen, er det smartere at funktionen tager disse to ting som argumenter. På den måde kan vi bruge samme funktion til at teste hvert spørgsmål og hvert svar. Funktionen skal også tage listen med de studerende som argument.**

Jeg har ændret funktionen til at indtage en specifikt svarmulighed, et specifikt spørgsmål og en liste med spørgeskemasvar som argumenter. Vi kan dermed anvende funktionen til alle spørgsmål og svarmuligheder.

### **2.3.2 (Percentage function:) Igen har du hardcoded listen med de studerende og at vi gerne vil kigge på spørgsmål 4. I stedet for at gøre dette, er det smartere, hvis funktionen tager spørgsmålsnummeret som argument, hvilket gør, at vi nemt kan anvende funktionen på flere spørgsmål.**

Jeg har i den rettede assignment redigeret funktionen til at indtage vores liste med surveysvar som et argument, og jeg har gjort det muligt, at indtage et specifikt spørgsmål som argument, så vi kan benytte funktionen til hvilket som helst spørgsmål fra survey'et.

Jeg forsøgte mit yderste på, at skildre vores printfn fra resten af funktionen, men kunne ikke finde en måde, hvor det virkede sammen med vores recursion.

### **2.3.3 (Same answer function:) Igen har du hardcoded listen med studerende og nummeret på spørgsmålet.**

Dette er nu rettet, så man kan anvende listen af surveysvar som argument, og spørgsmålet er ikke længere en angivet del af funktionen.

### **2.3.4 (Same answer function:) Denne funktion skulle egentligt returnere id's på to studerende med præcis de samme svarmuligheder, dvs. samme svar i alle spørgsmål. Overvej om du kan omskrive funktionen til at gøre dette.**

Funktionen tjekker nu for studerende, der har angivet samme svarmuligheder i alle spørgsmål, og returnerer id'en af disse.

### **2.3.5 Vi vil helst ikke mixe actions og computations. Dette gør du når du har print-statements i dine funktioner. Prøv om du kan ændre dette.**

Jeg har forsøgt på, at ændre funktionerne således, at der ikke er print statements i funktionernes let bindings, men har endnu ikke fundet en måde hvorpå, at man kan printe outputs fra en recursive funktion, uden at gøre det inde i funktionen.

## **2.4 Feedback on question 3**

### **2.4.1 Giv gerne et eksempel på en funktion**

Jeg har nu angivet et simpelt eksempel på en funktion i delen, hvor jeg forklarer, hvad en funktion er.

**2.4.2 (Features of functional programming:) Kan du uddybe lidt, og måske nævne andre konkrete fordele?**

Jeg har tilføjet, at i F# er alt en expression, hvilket gør det muligt, at skrive ens kode meget kompakt.

**2.4.3 Hvor specifikt i øvelserne har du oplevet fordele ved funktionel programmering?**

Jeg har nu nævnt i et afsnit, at vi blandt andet kan benytte pattern matching til at reducere behovet for if-else-kæder, hvilket gør koden mere klar i min mening.