

Functioneel Programmeren – opgaven practicum 1

Opgave 1. Introductie-opgave. Definieer de volgende functie

$$f\ x = 2x^2 + 3x - 5$$

en evalueer een paar keer voor een paar waarden van x .

Opmerking. Vanaf nu moet u bij elke functie die u definieert vooraf het type opgeven.

Opgave 2. De volgende functies zijn beschikbaar:

```
ord :: Char -> Number
chr :: Number -> Char
```

die naar/van de code (Unicode) van een character vertalen.

- Schrijf mbv `ord` en `chr` een functie `codeer` die een gegeven (hoofd)letter verandert in de letter die drie posities verderop in het alfabet staat (cyclisch, dus voorbij de 'z' verder gaan met de 'a'). Bijvoorbeeld

```
codeer 'a' = 'd'
codeer 'P' = 'S'
codeer 'y' = 'b'
```

De functie `codeer` moet alle overige characters (cijfers, leestekens, etc.) onveranderd laten.

Hint: de relaties $<$, \leq , \geq , $>$ werken ook op characters.

- Evalueer de expressies

```
map codeer "hallo"
map codeer "Morgenavond, 8 uur in Scheveningen"
```

(`map` zorgt ervoor dat `codeer` op alle letters van de strings wordt toegepast).

- Generaliseer uw functie `codeer` zodanig dat als extra argument een getal meegegeven kan worden dat aangeeft over hoeveel posities de te coderen letter verschoven moet worden (hierboven is dat aantal dus drie). Ga na in welke volgorde de argumenten meegegeven moeten worden om met behulp van `map` een string te kunnen coderen.

Opgave 3. Schrijf een *recursieve* functie die uitrekent hoeveel geld je na n jaren hebt als je begint met een bedrag b , en r procent rente per jaar krijgt. Hou rekening met rente over rente (die hoeft maar één keer per jaar te worden berekend).

Opgave 4. Schrijf twee functies `wortel1` en `wortel2` die (gegeven a, b, c) de wortels bepalen van de vierkantsvergelijking $ax^2 + bx + c = 0$ (neem aan dat $a \neq 0$). Geef als waarde van deze functies bij negatieve discriminant:

`error "discriminant negatief".`

Gebruik hierbij een ook zelf te schrijven aparte functie `discr` om de discriminant te bepalen.

Test uw functies voor een aantal waarden van a, b, c .

Opgave 5. Een tweedegraads polynoom is van de vorm $ax^2 + bx + c$ (neem weer aan dat $a \neq 0$).

- Schrijf een functie `extrX` die bij gegeven a, b, c bepaalt bij welke waarde van x deze polynoom zijn extreme waarde heeft.
- Schrijf een functie `extrY` die deze extreme waarde van de polynoom bepaalt.

Opgave 6. Schrijf recursieve definities voor de volgende lijstfuncties, en vermeld van elke functie het type.

- `mylength` die de lengte van een lijst oplevert,
- `mysum` die de elementen in een lijst van getallen optelt,
- `myreverse` die een lijst achterstevoren zet,
- `mytake` die de eerste n elementen uit een lijst oplevert (als n groter is dan de lengte van de lijst, moet de hele lijst worden opgeleverd),
- `myelem` die zegt of een gegeven element in een lijst zit,
- `myconcat` die een lijst van lijsten aan elkaar “plakt” tot één lange lijst,
- `mymaximum` die het maximum element uit een lijst oplevert,
- `myzip` die een paar van lijsten omzet in een lijst van paren (de kortste lijst is bepalend voor de lengte van de resultaatlijst).

Opgave 7. Een rij getallen is *rekenkundig* als (uitgaande van een zeker startgetal) elk volgend getal uit het voorgaande getal wordt verkregen door er een vast verschilgetal bij op te tellen. In de onderstaande opgaven wordt er steeds van uit gegaan dat er een rekenkundige rij is gegeven door startgetal **s** en verschil **v**.

- Schrijf een recursieve functie **r** die de bijbehorende rekenkundige rij genereert. Het type van **r** is dus: **r :: Number -> Number -> [Number]**.
- Schrijf een functie **r1** die het **n**-de getal uit die rij selecteert (het is handig om **r** te gebruiken).
- Laat **i** en **j** twee indices zijn (bedenk dat het eerste element uit een lijst index 0 heeft). Schrijf een functie **totaal** die de som bepaalt van het **i**-de element t/m het **j**-de element uit diezelfde rij (gebruik **r**, **take**, en **drop**).

Opgave 8.

- Schrijf een functie **allEqual** die nagaat of alle elementen van een lijst gelijk zijn.
- Schrijf een functie **isRR** die test of een lijst van getallen een rekenkundige rij is (d.w.z. het verschil tussen elk tweetal opeenvolgende getallen in de lijst is steeds hetzelfde).

Opgave 9. Een matrix kan worden gezien als een lijst van lijsten van getallen, waarbij de binnenste lijsten de rijen weergeven. Schrijf recursieve functies die:

- van een matrix test of alle rijen even lang zijn,
- van een matrix de lijst van totalen van alle rijen oplevert,
- een matrix transposeert, dus elke n^e rij wordt de n^e kolom,
- van een matrix de lijst van totalen van alle kolommen oplevert.