

TW1090 Inleiding Programmeren
Tussentoets
13 december 2016, 13:30 – 16:30 uur

Normering: Opgave 1 en 2: ieder 6 punten. Opgave 3: 4 punten. Opgave 4: 2 punten.
Score: totaal aantal punten + 2. Cijfer: score / 2

Voorbereiding

Lees eerst de instructies op het aparte Instructie blad (Inloggen, IDLE starten, pdf van het boek openen).

Gebruik bij de implementatie van de opgaven alleen de modules / bibliotheken die nu al door middel van imports worden geïmporteerd. Voeg dus geen import statements toe. Verander ook de hoofdprogramma's niet, indien al een hoofdprogramma is gegeven. Voeg dan alleen implementaties toe van de gevraagde functies en test deze functies in het hoofdprogramma. Vul je naam en studienummer ook in het commentaar van de programma's in (zie de inhoud van de .py files). Open in de IDLE omgeving de .py file die je wilt gaan editen. Zorg ervoor dat je steeds maar 1 .py file tegelijk geopend hebt. Save je .py files regelmatig.

Opgave 1 (1 + 2 + 3 punten)

Implementeer de functies **f(x)**, **a(n)** en **kdif(eps)** in het testprogramma **A1.py** en test je functies met het testprogramma.

a) Implementeer de functie $f(x) = \frac{3x+1}{6}$ met domein $\mathbb{R}$.

b) Gegeven is de rij $\{a_n\}_{n=1}^{\infty}$ gedefinieerd door de recurrente betrekking $a_n = \frac{3a_{n-1}+1}{6}$ met $a_1 = 1$

De eerste getallen van de rij zijn derhalve $1, \frac{2}{3}, \frac{1}{2}, \frac{5}{12}, \frac{3}{8}, \dots$

Implementeer de functie **a(n)** die het n-de getal uit de rij teruggeeft.

c) Implementeer een functie **kdif(eps)** die de kleinste waarde van k bepaalt waarvoor $a_k - a_{k+1} < \text{eps}$ geldt. De functie geeft de waarde van k en het verschil $a_k - a_{k+1}$ terug.

Hieronder zie je de gewenste uitvoer van het programma na implementatie van de drie functies

>>>

n	f(n)
1	0.6666666667
2	1.1666666667
3	1.6666666667
4	2.1666666667
5	2.6666666667

n	a(n)
1	1.00000000000
2	0.66666666667
3	0.50000000000
4	0.41666666667
5	0.37500000000
6	0.35416666667
7	0.34375000000
8	0.33854166667
9	0.33593750000
10	0.33463541667
11	0.33398437500
12	0.33365885420
13	0.33349609380
14	0.33341471350
15	0.33337402340

eps	k	a(k) - a(k+1)
1e-01	3	8.333333e-02
1e-02	7	5.208333e-03
1e-03	10	6.510417e-04
1e-04	13	8.138021e-05
1e-05	17	5.086263e-06
1e-06	20	6.357829e-07
1e-07	23	7.947286e-08
1e-08	26	9.934107e-09
1e-09	30	6.208817e-10
1e-10	33	7.761020e-11
1e-11	36	9.701295e-12
1e-12	40	6.063483e-13
1e-13	43	7.577272e-14
1e-14	46	9.492407e-15
1e-15	50	6.106227e-16

Push ENTER:

>>>

Opgave 2 (2 + 4 punten)

Implementeer de functies **conv(c)** en **convStr(cStr)** in het testprogramma **A2.py** en test je functies met het testprogramma.

- a) Implementeer een functie **conv(c)** die een karakter **c** converteert naar een ander karakter op de volgende manier:
- Het teken '*' wordt omgezet naar een spatie.
 - Een cijfer in '0' t/m '9' wordt omgezet naar een letter in 'a' t/m 'j'
 - Andere tekens komen niet in de invoer voor.

- b) Implementeer een functie **convStr(cStr)** die een string van karakters converteert naar een nieuwe string van karakters. Dit is een uitbreiding van de functie **conv(c)**, die je al hebt geïmplementeerd. Die functie **conv(c)** kun je daarom kopiëren naar **convStr(cStr)**.

Een string moet op de volgende manier worden geconverteerd:

- '*' moet worden omgezet naar een spatie
- '0' t/m '9' moet worden omgezet naar 'a' t/m 'j'
- '+0' t/m '+9' moet worden omgezet naar 'k' t/m 't'
- '++0' t/m '++5' moet worden omgezet naar 'u' t/m 'z'

In overzichtelijker vorm:

De tekens:	*	0123456789	+0123456789	++012345
worden omgezet naar:		abcdefghijkl	mnopqrst	uvwxyz

Voorbeeld:

Invoerstring cStr: "6+443*++5+4*74+9*+189+0+9*+94*+0+1+4+5+54+3"
g o e d z o h e t l i j k t t e k l o p p e n
Resultaat convStr(cStr): "goed zo het lijkt te kloppen"

Hieronder zie je de gewenste uitvoer van het testprogramma na implementatie van beide functies

```
>>>
>>>
conv('0') = a
conv('*') = 
conv('5') = f

convStr('0') = a
convStr('*') = 
convStr('5') = f
convStr('+5') = p
convStr('++5') = z
```

goed zo het lijkt te kloppen

Push ENTER:

```
>>> >>>
```

Opgave 3 (2 + 2 punten)

Implementeer een functie **evaluate(lst)** en een **recursieve** functie **count(lst)** in het testprogramma **A3.py** en test de functies met het testprogramma.

Er wordt in het testprogramma gebruik gemaakt van 3 lijsten:

data0: een (niet geneste) lijst met gehele getallen

[2, 3, 5, 8, 1, 6]

data1: een geneste lijst met de getallen 0, 1 en 2

[0, 1, 0, 0, [], [[2]], [0, 0, 2], [2, 2, 2, 2, 2, 1], 1, [], 1, 1, 1]

data2: een geneste lijst met de strings **""**, **"**, **"** en **"""**

['', ' ', '""', '', [' ', ' ', '""', [' ', ' ', ' ', ' ', '""']], [[[]]], ['"', ''], '']

- a) Implementeer een functie **evaluate(lst)** die een berekening uitvoert met de getallen in een (niet geneste) lijst. Het resultaat wordt berekend op volgende manier:
start met **res** gelijk aan 0
Indien het volgende getal in de lijst
- een drievoud is, dan wordt **res** met dat drievoud vermenigvuldigd
 - geen drievoud is maar wel een even getal, dan wordt het getal opgeteld bij **res**
 - geen drievoud is maar wel een oneven getal, dan wordt het getal afgetrokken van **res**

Voorbeeld:

Voor lijst **[2, 3, 5, 8, 1, 6]** wordt het resultaat van berekening $(((((0 + 2) * 3) - 5) + 8) - 1) * 6 = 48$

- b) Implementeer een functie **total(lst)** die in een geneste lijst **lst** het totaal aantal elementen telt dat niet gelijk is aan een lijst, dus bijvoorbeeld in lijst **data1** het aantal getallen.

Hints bij b):

- Gebruik recursie
- Gebruik **type(elt) == list** om te bepalen of **elt** een lijst of iets anders is.

Hieronder zie je de gewenste uitvoer van het testprogramma na implementatie van beide functies:

```
>>>
Data0 is:
[2, 3, 5, 8, 1, 6]
Data1 is:
[0, 1, 0, 0, [], [[2]], [0, 0, 2], [2, 2, 2, 2, 2, 1], 1, [], 1, 1, 1]
Data2 is:
[' ', ' ', '""', '', [' ', ' ', '""', [' ', ' ', ' ', ' ', '""']], [[[]]], ['"', ''], '']
```

Evaluate data0: 48 (zie de berekening in het voorbeeld bij onderdeel a)

Total data1: 18 (namelijk in totaal 18 niet lijst elementen in lijst data1)

Total data2: 13 (namelijk in totaal 13 niet lijst elementen in lijst data2)

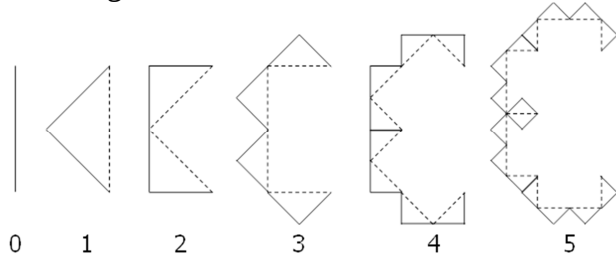
Push ENTER:

```
>>>
```

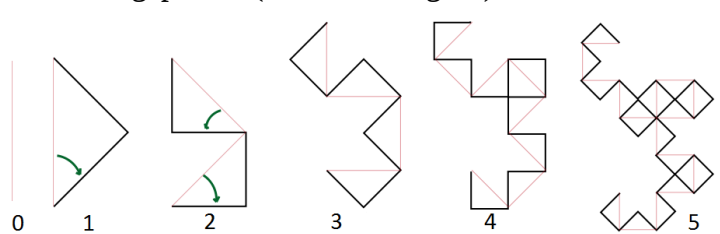
Opgave 4 (2 punten)

Een C-kromme wordt verkregen door het volgende proces. Start met 1 lijnstuk. Vervang voor iedere volgende generatie alle lijnstukken door twee nieuw lijnstukken die aan de linkerkant van de kromme een rechte hoek vormen. Doe dit bij iedere generatie voor alle lijnstukken aan dezelfde kant van de kromme.

Bij een drakenkromme worden de vervangende twee lijnstukken die een rechte hoek vormen bij het tekenen van één generatie afwisselend rechts en links van de kromme geplaatst (zie rechter figuur).

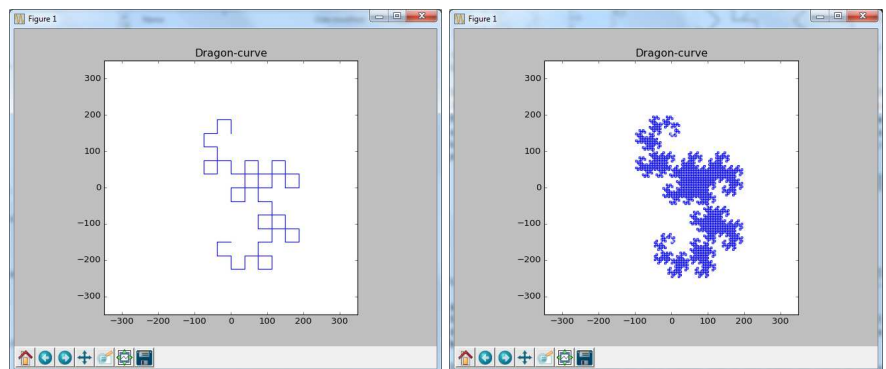


Generatie 0 t/m 5 van een C-kromme



Generatie 0 t/m 5 van een drakenkromme

Het testprogramma **A4.py** beeldt in de toestand waarin je het bestand aantreft een C-kromme af. Probeer dit maar uit door het programma te runnen. De functie in het programma is alvast **DrawDragon** genoemd. Pas het programma met de functie **DrawDragon(direction, length, n)** aan zodat in plaats van een C-kromme een Drakenkromme wordt afgebeeld. Naast de functie header en inhoud mag ook het hoofdprogramma indien nodig worden aangepast. Er zijn slechts enkele kleine aanpassingen nodig om het programma een drakenkromme te laten afbeelden in plaats van een C-kromme.



Voorbeeld resultaat (generatie 6 en 12):

Afsluiten

Sluit alle .py bestanden. Controleer in IDLE door middel van Kies File | Open nog een keer extra of je in je directory **Homedrive(H:)** vier .py bestanden hebt staan met in de naam jouw naam en studienummer. **Deze bestanden vormen het werk dat je inlevert.**

Klik nu op het sluihtokje rechts bovenin het IDLE window. Sluit op deze manier alle Editor en Python Shell windows van IDLE. Sluit ook de pdf van het boek. Kies tenslotte Start | Exit | OK. Je wordt nu uitgelogd. Dit duurt vrij lang.

AAN DE HAND VAN UITSLUITEND DE .PY BESTANDEN WORDT JE WERK BEOORDEELD. INDIEN DE .PY BESTANDEN NIET OP HOMEDRIVE(H:) STAAN MAAR OP DE DESKTOP, DAN HEB JE GEEN WERK INGELEVERD.

einde opgaven