

TW1090 Inleiding Programmeren
Tussentoets
8 december 2015, 13:30 – 16:30 uur

Normering: Opgave 1 t/m 3 ieder 6 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

De reeks $\sum_{n=0}^k (-1)^n \frac{x^{n+3}}{2^{n+1}}$ is op $(-2, 2)$ absoluut convergent en heeft als som $f(x) = \frac{x^3}{x+2}$. Implementeer in

het programma **A1.py** een functie **PartieleSom(x, k)** die $f(x)$ benadert met de partiële som

$\sum_{n=0}^k (-1)^n \frac{x^{n+3}}{2^{n+1}}$. Implementeer tevens een functie die $\frac{x^3}{x+2}$ bepaalt. Implementeer ook een hoofdprogramma

dat werkt zoals in de voorbeelden op de volgende bladzijde. Voor het berekenen van de resultaten moeten in het hoofdprogramma de geïmplementeerde functies worden aangeroepen. In het hoofdprogramma moet eerst drie keer de functie f worden aangeroepen met als argument respectievelijk de gehele getallen -1, 0 en 1. Vervolgens moet gevraagd worden om waarden voor k en x (tussen -2 en 2). Tenslotte moeten de resultaten $f(x)$ en $\text{PartieleSom}(x, k)$ worden getoond. Als je programma niet werkt voor het 3^e geval: $k=10000$ en $x=-1.99$ dan krijg je maximaal 5 van de 6 punten. Zodra je een versie hebt die werkt voor de eerste twee gevallen, maak dan een kopie van de functie **PartieleSom**, commentarieer een van de twee functies **PartieleSom** uit (code selecteren en Format | Comment Out Region kiezen) en probeer de andere versie zo uit te breiden dat het 3^e geval ook werkt. Zie hieronder de gevraagde uitvoer voor invoer resp. $k=10$ en $x=1.2$, $k=100$ en $x=-1.5$ en $k=10000$ en $x=-1.99$

```
>>>
f(-1) = -1.0
f( 0) = 0.0
f( 1) = 0.333333333333
```

```
Enter k: 10
-2 < x < 2
Enter x: 1.2
```

```

                f(x) = 0.54
PartieleSom(x, k) = 0.541959104102
>>> ===== RESTART =====
>>>
f(-1) = -1.0
f( 0) = 0.0
f( 1) = 0.333333333333

Enter k: 100
-2 < x < 2
Enter x: -1.5

                f(x) = -6.75
PartieleSom(x, k) = -6.75
>>> ===== RESTART =====
>>>
f(-1) = -1.0
f( 0) = 0.0
f( 1) = 0.333333333333

Enter k: 10000
-2 < x < 2
Enter x: -1.99

                f(x) = -788.0599
PartieleSom(x, k) = -788.0599
>>>

```

Opgave 2

Implementeer in het programma **A2.py** een functie **SinCosTan(n)** die een lijst van lijsten teruggeeft. Ieder lijstelement is een lijst met een sinus, een cosinus en een tangens van een hoek. De hoeken voor de opeenvolgende lijstelementen zijn regelmatig verdeeld over het interval $[0, 2\pi]$. Zie onderstaand voorbeeld voor $n = 3$.

```
[[sin(0), cos(0), tan(0)], [sin( $\frac{2}{3}\pi$ ), cos( $\frac{2}{3}\pi$ ), tan( $\frac{2}{3}\pi$ )], [sin( $\frac{4}{3}\pi$ ), cos( $\frac{4}{3}\pi$ ), tan( $\frac{4}{3}\pi$ )], [sin( $2\pi$ ), cos( $2\pi$ ), tan( $2\pi$ )]]
```

Implementeer ook een functie **PrintLijst(n)** die eerst de hierboven besproken lijst van lijsten maakt en deze vervolgens geformatteerd afdruckt. De uitvoer moet er precies zo uitzien als in de voorbeelden op de volgende bladzijde.

Wederom krijg je maximaal 5 van de 6 punten als het 3^e voorbeeld niet goed werkt. Hier moeten de streepjes in de tabel komen bij $\tan(\frac{\pi}{2})$ en $\tan(\frac{3\pi}{2})$.

```

>>>
Enter a value for n: 3

  *pi   sin   cos   tan
0/ 3   0.00   1.00   0.00
2/ 3   0.87  -0.50  -1.73
4/ 3  -0.87  -0.50   1.73
6/ 3  -0.00   1.00  -0.00
>>> ===== RESTART =====
>>>
Enter a value for n: 15

```

```

    *pi    sin    cos    tan
0/15    0.00    1.00    0.00
2/15    0.41    0.91    0.45
4/15    0.74    0.67    1.11
6/15    0.95    0.31    3.08
8/15    0.99   -0.10   -9.51
10/15   0.87   -0.50   -1.73
12/15   0.59   -0.81   -0.73
14/15   0.21   -0.98   -0.21
16/15  -0.21   -0.98    0.21
18/15  -0.59   -0.81    0.73
20/15  -0.87   -0.50    1.73
22/15  -0.99   -0.10    9.51
24/15  -0.95    0.31   -3.08
26/15  -0.74    0.67   -1.11
28/15  -0.41    0.91   -0.45
30/15  -0.00    1.00   -0.00
>>> ===== RESTART =====
>>>
Enter a value for n: 4

    *pi    sin    cos    tan
0/ 4    0.00    1.00    0.00
2/ 4    1.00    0.00    ----
4/ 4    0.00   -1.00   -0.00
6/ 4   -1.00   -0.00   ----
8/ 4   -0.00    1.00   -0.00
>>>

```

Opgave 3

In het testprogramma **A3.py** van deze opgave worden de lijsten gemaakt die op de volgende bladzijde zijn weergegeven. De lijsten worden bomen genoemd. In de (interne) knopen van de bomen bevinden zich letters. In de bladeren bevinden zich gehele getallen.

Implementeer een **recursieve** functie **TreeResult(tree)** die de letters uit de knopen in een string plaatst in de volgorde waarin ze worden tegengekomen bij het van links naar rechts afdalen door alle takken van de boom naar de bladeren. Tevens berekent de functie de som van alle even getallen min de som van alle oneven getallen in de bladeren. De beide resultaten, dat wil zeggen de string en de som, worden door de functie teruggegeven. De volgende lijsten (bomen) worden in het testprogramma gebruikt.

```

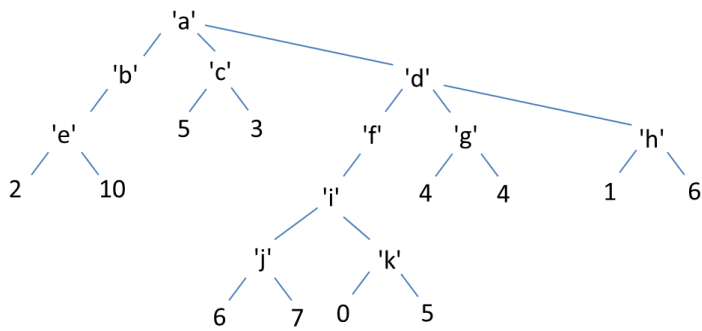
tree1 = ['a', ['b', 10, 8], ['c', 5, 2]]

tree2 = ['a']

tree3 = ['a',
        ['b', ['e', 2, 10]],
        ['c', 5, 3],
        ['d',
         ['f', ['i', ['j', 6, 7], ['k', 0, 5]]],
         ['g', 4, 4],
         ['h', 1, 6]
        ]
       ]

```

Hieronder wordt de boom **tree3** grafisch weergegeven:



Gewenste uitvoer van het testprogramma:

```
>>>
tree1:
abc
15

tree2:
a
0

tree3:
abecdfijkgh
11
```

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:)** drie .py bestanden hebt staan met in de naam jouw naam en studienummer. **Deze drie bestanden vormen het werk dat je inlevert.**

Druk nu op het sluihokje 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 DRIE .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