

Inleiding programmeren

1^e jaar wis-, natuur- en sterrenkunde

Universiteit van Amsterdam

Hertentamen 2020-2021 (blok 1)

2 december 2020, 09:00-11:30 uur

Dit tentamen bestaat uit 5 vragen. Er zijn 10 punten te verdelen en je moet minimaal zes punten hebben om te slagen.

Corona-examen: We nemen dit online tentamen af zonder proctoring, maar dit vertrouwen geeft natuurlijk ook een verantwoordelijkheid aan jullie kant. We vragen jullie om dit examen **zelfstandig** te maken. Zonder hulp van buitenaf, dus zonder onderling overleg of hulp via internet. Zoals gebruikelijk is er een plagiaat-check.

Regels:

- Maak alle opdrachten in de file `tentamen.py` en lever deze in.
- Schrijf je naam en studentnummer bovenaan je file.
- In elke opdracht schrijf je één functie; voor uitvoer van het berekende antwoord gebruik je in elke functie gewoon `print`-opdrachten.
- Je mag in dit tentamen geen functies uit de `numpy` bibliotheek gebruiken, tenzij anders aangegeven in een opdracht.
- Zorg er bij het inleveren van je code voor dat bij het runnen van je programma **alle** functies uitgevoerd worden. Dat helpt bij het nakijken.

Opgave 1: Lijsten manipuleren (2 punten)

Schrijf een functie `Opgave1(L)` die van een gegeven lijst L bekijkt welke getallen een veelvoud zijn van drie en vier en die opslaat in een lijst. Bekijk verder voor elk van de getallen die in die lijst terechtkomen of ze ook toevallig een veelvoud zijn van vijf. Sla de getallen die daar **niet** aan volden in een aparte lijst op en zorg dat aan het eind van je programma de twee lijsten geprint worden.

Gebruik het onderstaande voorbeeld om je functie te controleren. Als je in je programma de functie aanroept met de volgende input-lijst:

```
L = [10, 1299, 23, 12, 4, 99, 100, 36, 48, 120]
Opgave1(L)
```

dan moet er op het scherm de volgende output verschijnen:

```
Veelvouden van 3 en 4: [12, 36, 48, 120]
Veelvouden van 3 en 4 (maar niet van 5): [12, 36, 48]
```

Zorg dat je functie ook werkt met andere input-lijsten. Bij het nakijken wordt dat gecontroleerd.

Opgave 2: benaderen van π (2 punten)

Er bestaan verschillende wiskundige reeksen die het getal π benaderen:

$$\text{Leibniz : } \quad \pi = 4 \cdot \sum_{n=0}^{n=\infty} \frac{(-1)^n}{2n+1}$$

$$\text{Euler : } \quad \pi^2 = 6 \cdot \sum_{n=1}^{n=\infty} \frac{1}{n^2}$$

We kunnen met de computer de reeks niet tot $n = \infty$ evalueren, maar kunnen de reeks wel afkappen bij een willekeurig grote waarde N , met N een groot getal. Voor $N = 10$ krijgen we als schattingen: $\pi = 3.2323$ (Leibniz) en $\pi = 3.0494$ (Euler). De schatting van Leibniz benadert π het best en wijkt voor $N = 10$ maar 2.9% af.

Schrijf een functie `Opgave2()` die voor vier waarden van N ($N=10, 100, 1000, 10000$) bekijkt welke reeks π het dichtst benadert. Print naar het scherm zowel beide schattingen (op 5 decimalen), beide afwijkingen in procent (op 5 decimalen) als de naam van de wiskundige wiens benadering π het dichtst nadert.

Als output op je scherm verschijnt dus:

```

n = 10: Leibniz = 3.23232 (delta = 2.88781 procent)
n = 10: Euler = 3.04936 (delta = 2.93580 percent)
n = 10: Leibniz wint

n = 100: Leibniz = ...

```

In deze opgave is het toegestaan om de functie `sqrt()` te gebruiken. Zorg dat je programma dus begint met: `import math`

Opgave 3: Wiskundige eigenaardigheden (1 punt)

Voor elk paar a en b ($a, b \in \mathbb{N}$ en $a, b > 0$) is de ratio f gedefiniëerd als:

$$f = \frac{a^2 + b^2}{(ab + 1)}$$

Het blijkt dat als f een geheel getal is (dus als $(ab + 1)$ een perfecte deler is van $(a^2 + b^2)$), dat f dan ook altijd een kwadraat is: $f = n^2$, met $n \in \mathbb{N}$.

Schrijf een functie `Opgave3()` die alle combinaties van a en b vindt waarbij f een geheel getal is. Beperk je tot $1 \leq a \leq 1000$ en $1 \leq b \leq 1000$ en hou ook bij hoeveel paren je vindt.

Print alle combinaties op het scherm en print aan het eind van je programma ook hoeveel paren je in totaal hebt gevonden. Let op: er mogen *géén* paren dubbel voorkomen in je lijst. Dus niet (a,b) is zowel $(2,8)$ en $(8,2)$, maar alleen de eerste.

Gebruik de volgende syntax voor de output van de paren en het totaal aantal:

```

a = xxx b = xxx -> f = xxx    (dit is xxx kwadraat)
a = yyy b = yyy -> f = yyy    (dit is yyy kwadraat)
...

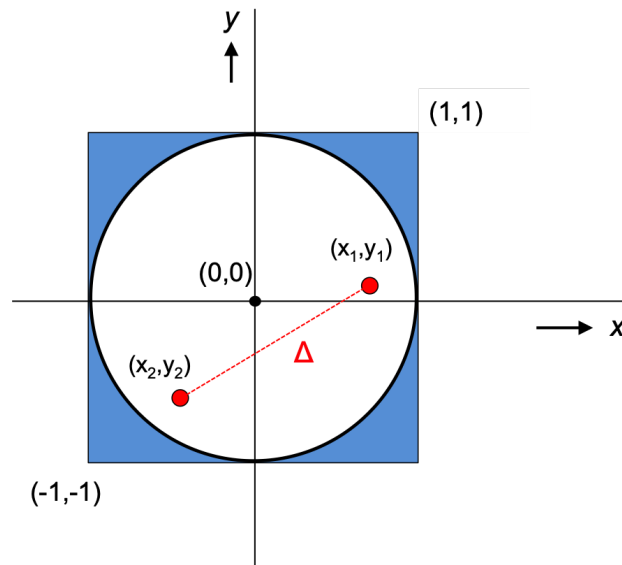
```

Er zijn in totaal zzz paren gevonden

Tip: - het is handig om gebruik te maken van de modulo operator (%) in Python.
 - Je mag in deze opdracht gebruik maken van `math.sqrt()`

Opgave 4: Gemiddelde afstand tussen 2 punten in een cirkel (2 punten)

Schrijf een functie `Opgave4()` die de gemiddelde afstand bepaalt tussen twee punten in een cirkel met straal 2.



Gebruik hierbij de volgende strategie:

Genereer heel vaak twee random punten (x_1, y_1) en (x_2, y_2) in een vierkant met een zijde met lengte 2: $(-1 \leq x \leq 1)$ en $(-1 \leq y \leq 1)$. Bepaal voor elk paar punten steeds eerst of **beide punten** in de cirkel liggen. Als dat zo is, doe dan ook de volgende stappen voor deze zogenaamde 'in-de-cirkel' paren:

- 1) Bereken de onderlinge afstand (Δ)
- 2) Bewaar de totale afstand (som van alle afstanden) voor deze 'in-de-cirkel' paren
- 3) Bereken de gemiddelde afstand door de totale afstand te delen door het totaal aantal 'in-de-cirkel' paren

Aan het eind van je programma moet de gemiddelde afstand (met 2 decimalen nauwkeurigheid) op het scherm geprint worden:

De gemiddelde afstand tussen 2 punten in een cirkel = x.xx

Om random getallen te gebruiken heb je de `random()` functie nodig. Zorg dat je programma dus begint met: `from random import *`

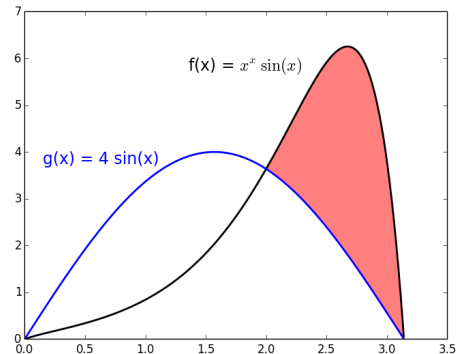
Opgave 5: Numeriek integreren (2 punten)

Schrijf een functie `Opgave5()` die de oppervlakte berekend van het (rood-gearceerde) gebied dat ingeklemd is tussen de twee functies $f(x) = x^x \sin(x)$ en $g(x) = 4 \sin(x)$ met behulp van de Monte Carlo-methode. Zorg hierbij dat je antwoord op 2 decimalen nauwkeurig weergegeven wordt.

In deze opgave bekijken we de functies:

$$\begin{aligned} f(x) &= x^x \sin(x) \\ g(x) &= 4 \sin(x) \end{aligned}$$

In de figuur hiernaast zijn de grafieken $f(x)$ en $g(x)$ weergegeven. Voor de volledigheid: het gaat hierbij om de oppervlakte van het rode gebied.



Ga bij het bepalen van de oppervlakte als volgt te werk:

- Gooi een groot aantal ($N=100000$) random punten in een gebied van bekende grootte rond het integratiegebied. Voor elk punt (x_i, y_i) trek je dus 2 random getallen: x_i en y_i .
- Bepaal de fractie van het aantal punten dat in het rode gebied valt
- Bereken uit deze fractie de oppervlakte tussen de twee grafieken

Als output op je scherm verschijnt dan (met 2 decimalen):

De oppervlakte van het rode gebied is x.xx

Tip: maak een grafiek van de 'goede' punten. Liggen ze echt in het goede gebied?

Benodigde bibliotheken: om random getallen te gebruiken heb je de `random()` functie nodig. Zorg dat je programma dus begint met: `import random`. De functie `sin()` functie vinden in de wiskundebibliotheek. Schrijf bovenaan je programma dus ook: `import math`.