Algemeen

Oneindige lussen

Let op voor oneindige lussen. Een oneindige lus is een lus die eindeloos blijft uitgevoerd worden: in de meeste gevallen gaat het om een `while`-lus waarbij de statements binnen de lus er nooit voor zorgen dat de voorwaarde uiteindelijk `False` wordt. Bekijk als bijvoorbeeld eens het volgend stuk code

```
>>> i = 0
>>> a = 0
>>> while i < 4:
...     a += 1
```

Omdat het statement `a += 1` er nooit zal voor zorgen dat de waarde van de variabele `i` groter dan of gelijk aan 4 wordt, zal de voorwaarde `i < 4` altijd de waarde `True` opleveren.

Tip: Als je werkt met Pycharm, dan kan je nagaan dat je programma aan het uitvoeren is, door het rode vierkantje links van de Console of rechtsboven in de menubalk. Als je op dit vierkantje klikt, dan forceer je om de uitvoering van het programma te stoppen.

Lussen vroegtijdig afbreken

In Python kan je gebruikmaken van de statements `break` en `continue` om een lus vroegtijdig af te breken. Deze statements worden echter algemeen aanzien als slechte programmeerstijl. Je kunt ze dus beter niet gebruiken, want ze zullen je punten kosten als je ze gebruikt op een evaluatie of een exam.

Een situatie waarin je een lus vroegtijdig zou willen afbreken, doet zich bijvoorbeeld voor als je op zoek moet gaan naar een oplossing door een aantal mogelijke gevallen uit te proberen, en te stoppen van zodra je één oplossing gevonden hebt. In plaats van te werken met `break` en `continue` kan je in dit geval beter werken met een variabele die aangeeft of de oplossing al gevonden werd

```
>>> gevonden = False
>>> while not gevonden:
...     if <oplossing gevonden>:  # <oplossing gevonden> stelt voorwaarde voor
...         gevonden = True
...
```

Van zodra de oplossing gevonden werd (hier aangegeven door het feit dat de voorwaarde *oplossing gevonden* de waarde `True` aanneemt), wordt de variabele `gevonden` op `True` gezet, waardoor uit de `while`-lus zal gesprongen wanneer de `while`-voorwaarde opnieuw geëvalueerd wordt na de huidige iteratiestap.

Inlezen van meerdere regels

Als je vooraf weet hoeveel regels er moeten ingelezen worden, dan kan je gebruikmaken van de volgende constructie

```
>>> aantal_regels = 4
>>> for _ in range(aantal_regels):
...     regel = input()
...     # doe iets met de regel
...
```

Als je niet op voorhand weet hoeveel regels er zijn, maar je weet dat de laatste regel bijvoorbeeld leeg is, dan kan je gebruikmaken van de volgende constructie

```
>>> regel = input()
>>> while regel:
...
...     # doe iets met de regel
...
```

```
...     regel = input()
...
```

Getallen naar boven afronden

De `math` module bevat een functie `ceil` die kan gebruikt worden om *floating point* getallen naar boven af te ronden. Deze functie geeft de afronding terug als een integer.

```
>>> import math
>>> math.ceil(3.2)
4
>>> math.ceil(3.7)
4
```

De `math` module bevat ook de omgekeerde functie `floor` die kan gebruikt worden om *floating point* getallen naar beneden af te ronden. Voor de klassieke manier om *floating point* getallen af te ronden kan je gebruikmaken van de ingebouwde functie `round`.

Canvascrack

Tellen vanaf nul

Informatici beginnen standaard te tellen vanaf 0, niet vanaf 1. Python volgt deze traditie op heel wat verschillende plaatsen. Zo genereert de ingebouwde functie `range` een reeks opeenvolgende getallen die begint bij 0, als je slechts één enkel argument meegeeft aan de functie. Zo tel je bijvoorbeeld standaard tot 5 in Python

```
>>> for i in range(6):
...     print(i)
...
0
1
2
3
4
5
```

Als je niet wil beginnen tellen vanaf 0, dan kan je de startwaarde meegeven als een tweede argument aan de `range` functie.

```
>>> for i in range(1, 6):
...     print(i)
...
1
2
3
4
5
```

Het wordt echter als meer *pythonic* beschouwd om het voorgaande op de volgende manier uit te schrijven

```
>>> for i in range(5):
...     print(i + 1)
...
1
2
3
```

```
4
5
```

Boeketje rozen

Specifieke info

Om dit probleem op te lossen overloop je best alle mogelijke hoeveelheden witte rozen. Met dit aantal kan je dan het bijhorende aantal rode en blauwe rozen berekenen. Wanneer de som van het aantal rode en het aantal blauwe rozen voldoet aan de opgegeven voorwaarde, dan print je het resultaat uit.