

Hierna volgen enkele oefeningen op het gebruik van lijsten, verzamelingen en woordenboeken in Python.

1. Schrijf een programma dat het opgooien van twee eerlijke dobbelstenen simuleert. In totaal worden de twee dobbelstenen 1000 maal opgegooid en wordt telkens de som van het aantal ogen berekend. Het programma geeft een overzichtelijke frequentietabel met de relatieve frequenties van de som van het aantal ogen. Vergelijk deze frequenties met de theoretische frequenties die je verwacht op basis van kansrekening.
2. Schrijf een programma dat op een willekeurige manier steeds exact 100 *verschillende* gehele getallen genereert gelegen tussen 0 en 1000. De getrokken getallen worden in een overzichtelijke tabel getoond. Verder worden ook het minimum, maximum en mediaan van die getallen berekend en getoond. De mediaan is de “middelste” waarneming wanneer die gesorteerd zijn van klein naar groot. Indien de “middelste” waarneming niet bestaat, is de mediaan het rekenkundig gemiddelde van de twee “middelste” waarnemingen.
3. Een echte deler van een natuurlijk getal is een (positieve) deler verschillend van het getal zelf. Dus, het getal 6 heeft als echte delers 1, 2 en 3. Een natuurlijk getal noemt men perfect indien het gelijk is aan de som van zijn echte delers. Het is duidelijk dat 6 een perfect natuurlijk getal is want $1 + 2 + 3 = 6$.

Schrijf een programma dat alle perfecte getallen tussen 1 en 10000 berekent. Die getallen worden niet enkel getoond maar ook de som die moet uitgerekend worden om de perfectheid te controleren. Bijvoorbeeld, bij het getal 6 wordt vermeld dat $1 + 2 + 3 = 6$.

4. Schrijf een programma dat eerst vraagt naar de invoer van een zin. In die zin worden vervolgens de verschillende gebruikte letters geïdentificeerd. Hierbij wordt geen onderscheid gemaakt tussen drukletters en kleine letters. Spaties, leestekens en andere niet-letters worden genegeerd. Die unieke gebruikte letters worden aan de gebruiker in alfabetische volgorde getoond. Tevens wordt ook getoond welke letters uit het alfabet niet zijn gebruikt in de opgegeven zin. Deze operatie wordt herhaald tot er als zin één of meerdere spaties worden opgegeven.
5. Een zin is een anagram van een tweede zin indien beide zinnen precies dezelfde letters bevatten. Hierbij wordt geen onderscheid gemaakt tussen kleine letters en hoofdletters en worden leestekens en spaties genegeerd. Zo is bijvoorbeeld “Mr Mojo Risin” een anagram van “Jim Morrison” (zanger van The Doors).

Schrijf een programma dat naar twee zinnen vraagt en controleert of de ene een anagram is van de andere. Het resultaat wordt getoond op het scherm waarna gevraagd wordt of er nog andere zinnen moeten worden gecontroleerd.

6. In het bordspel Scrabble hebben letters een bepaalde letterwaarde. In Wikipedia vindt men de letterwaarde voor de letters in het Nederlandse alfabet. De woordwaarde van een gegeven woord wordt bepaald door de som te maken van de letterwaarden van alle letters die voorkomen in dat woord.

Schrijf een programma dat de woordwaarde berekent van een opgegeven woord. Het resultaat wordt getoond. Dit proces wordt herhaald tot er één of meerdere spaties worden opgegeven als woord.

7. Schrijf een programma dat de natuurlijke getallen van 0 tot en met 999 voluit kan schrijven. Het programma vraagt naar een getal en geeft vervolgens de geschreven versie. Dus, bij een invoer van 325 hoort de weergave “driehonderdvijfentwintig” te verschijnen. Het programma blijft vragen naar nieuwe invoer tot “einde” wordt opgegeven.

8. Bij slecht verstaanbare mondelinge communicatie kan soms gevraagd worden om woorden of codes te spellen. Het spellingsalfabet van de NAVO is één van de hulpmiddelen die dan kan gebruikt worden om tekens in een woord of code te spellen. Zo wordt 205ABZ gespeld als “two zero five alfa bravo zulu”. Het volledige spellingsalfabet van de NAVO is terug te vinden op Wikipedia.

Schrijf een programma dat een willekeurige code of woord kan omzetten naar de geschreven spelling volgens het spellingsalfabet van de NAVO. Het programma blijft vragen naar nieuwe invoer tot één of meerdere spaties worden opgegeven.

9. Schrijf een programma dat een steekproef genereert van n observaties uit een normaal verdeelde populatie met gemiddelde μ en standaardafwijking σ . Vervolgens wordt het bereik van die steekproef (dit is het interval van de kleinste observatie tot de grootste observatie) in m gelijke klassen verdeeld en wordt er geteld hoeveel observaties voorkomen in elke klasse. Het resultaat van die telling wordt in een overzichtelijke tabel getoond. De noodzakelijke waarden n , m , μ en σ worden uiteraard eerst aan de gebruiker opgevraagd.
10. Schrijf een programma dat de verzameling berekent van alle mogelijke deelverzamelingen van een gegeven verzameling. Het handigste hierbij is de verzameling te beschouwen als een lijst. Het programma vraagt eerst naar alle elementen van een verzameling. Die elementen worden in één keer ingevoerd, gescheiden door een komma. Vervolgens worden alle mogelijke deelverzamelingen (eveneens in lijstvorm) bepaald en getoond. Stel bijvoorbeeld dat de letters a, b en c worden ingegeven als namen van de drie elementen van een verzameling. Dan wordt bijvoorbeeld de volgende output gegenereerd:

```
[ ]  
[ 'a' ]  
[ 'a', 'b' ]  
[ 'a', 'b', 'c' ]  
[ 'a', 'c' ]  
[ 'b' ]  
[ 'b', 'c' ]  
[ 'c' ]
```

Er zijn verschillende manieren denkbaar om het gewenste resultaat te bekomen. Eén mogelijkheid is gebaseerd op de bijtjie tussen de verzameling van alle deelverzamelingen van een gegeven verzameling A en de verzameling van alle binaire getallen met evenveel cijfers als het aantal elementen in A . Het cijfer 1 wordt toegekend indien het overeenkomstige element uit A voorkomt in de deelverzameling. Het cijfer 0 geeft aan dat het overeenkomstige element niet voorkomt in de deelverzameling. Hier volgen enkele voorbeelden voor $A = \{a, b, c\}$:

$$\begin{aligned}\emptyset &\leftrightarrow 000 \\ \{a, c\} &\leftrightarrow 101 \\ \{c\} &\leftrightarrow 001 \\ \{a, b, c\} &\leftrightarrow 111\end{aligned}$$