Oefeningen Python basics

Algemene opmerking:
Bij veel oefeningen nemen we input van de gebruiker aan. Een mooie uitdaging is om ervoor te zorgen dat je programma robuust is voor foutieve input. Dit betekent dat je programma merkt dat de gebruiker foutieve input geeft en dit signaleert aan de gebruiker (gebruik hiervoor *isdigit* om te controleren of een *string* kan omgezet worden naar een *integer*).

1. Vraag de gebruiker voor zijn leeftijd en geef als output het jaar waarin de gebruiker 100 jaar zal zijn.
    a. In eerste instantie mag je het huidige jaartal hard-coden in je code.
    b. Advanced: Zorg ervoor dat je programma volgend jaar ook nog werkt.

Opmerking: gebruik *leeftijd = input("<tekst>")* om input aan te nemen van de gebruiker.

2. Schrijf een programma dat de gebruiker vraagt om een getal.
    a. Geef als output weer of het getal even of oneven is.
    b. Geef als output weer of het getal even/oneven is; geef verder ook aan of het gegeven getal deelbaar is door 4.

3. Schrijf een programma dat vraagt om een getal *n* en als output de eerste *n* Fibonacci getallen geeft.
    Uitbreiding: schrijf een programma dat gegeven de parameters P & Q de eerste *n* Lucas getallen teruggeeft.
    $$\begin{aligned} U(0) &= 0 \\ U(1) &= 1 \\ U(n+2) &= PU(n+1) - QU(n) \end{aligned},$$
    Zie ook: https://en.wikipedia.org/wiki/Generalizations_of_Fibonacci_numbers

4. Schrijf een paswoord generator die, gegeven een getal *n* een paswoord genereert dat bestaat uit *n* willekeurige karakters.
    Opmerking: Bekijk zeker *random.sample*.

5. Schrijf een programma dat gegeven een string de string volledig omdraait (bvb Ik ben Tim wordt Tim ben ik).
    Opmerking: Gebruik *.split()*.

6. Schrijf een spel schaar-steen-papier waarbij je continu vraagt voor input van de gebruiker; je hebt dan 4 keuzes: "schaar", "steen", "papier" of "stop". Als de gebruiker stop ingeeft stopt het programma en bij de keuze van schaar/steen/papier speel je een spelletje schaar steen papier tegen de computer. Geef aan wie er gewonnen is en de tussenstand.
    Opmerking: gebruik *random.sample*.

a. Voor extra moeilijkheid: Zorg dat het programma robuust is voor meerdere talen/gewoontes (blad steen schaar & rock paper scissors).
b. Zorg dat het programma ook werkt met typfouten: het programma moet voor elke input de meest waarschijnlijke keuze nemen (tip: je kan gebruik maken van de Levenshtein distance)