

Semesteroppgave 2

INF620 - Høstsemesteret 2021

Semesteroppgavene er obligatoriske, og må være godkjente for å bestå INF620. Du skal levere én zip-fil, `sem2.zip`, som inneholder de 2 filene `oppg1.py`–`oppg2.py`. For å komprimere en eller flere filer til en zip-fil høyreklikker du filene (i dette tilfellet `oppg1.py`–`oppg2.py`) i maskinens filnavigasjonspprogram og velger **Komprimer** eller **Send til** → **Komprimert mappe**.

Frist: Torsdag 14. oktober kl 23:59

1 Navnelister

Svar leveres på fil med navn `oppg1.py` *Hint:* I denne oppgaven kan du få god bruk for streng-metoden `title`, som lager en kopi av strengen der hvert ord har stor forbokstav, men som forøvrig består av små bokstaver. Gjør deg kjent med `streng.title()` i konsollen før du løser oppgaven!

- a) (10%) Skriv en funksjon `les` som leser inn så mange personnavn (strenger) som brukeren ønsker å gi. Innlesing avsluttes med en tom streng. Funksjonen skal returnere en liste med de innleste navnene. Som parameter skal funksjonen ta en streng som sier hvilken kategori (f.eks. ‘gruppeleder’ eller ‘administrator’) personene tilhører. Eksempel på kjøring av funksjonen fra konsollen i Spyder:

```
In[1]: les('gruppeleder')
Første gruppeleder: sAndRa
Neste gruppeleder: maTHilde
Neste gruppeleder:
Out[1]: ['sAndRa', 'maTHilde']
```

- b) (10%) Skriv en funksjon `skriv_pent` som tar en liste med strenger som parameter, og som skriver ut strengene med stor forbokstav og med små bokstaver ellers. Eksempel på kjøring av funksjonen fra konsollen i Spyder:

```
In[2]: skriv_pent(['sAndRa', 'maTHilde'])
Sandra
Mathilde
```

- c) (10%) Skriv en funksjon `flett` som tar to lister (`liste1` og `liste2`) som parametre, og som returnerer en ny liste som består av elementene i `liste1` og `liste2`. I listen som returneres skal rekkefølgen være slik at elementene fra

`liste1` og `liste2` kommer vekselvis, dvs, `liste1[0]`, `liste2[0]`, `liste1[1]`, `liste2[1]`, ..., så langt det finnes element i begge listene. Eksempel på kjøring av funksjonen fra konsollen i Spyder:

```
In[3]: flett(['Dusan', 'Mathias', 'Eirik', 'Fredrik'],
           ['Mathilde', 'Sandra'])
Out[3]: ['Dusan', 'Mathilde', 'Mathias',
         'Sandra', 'Eirik', 'Fredrik']
```

- d) (10%) Skriv en funksjon `korrigjer` som tar en liste med strenger som param-eter, og som korrigjerer innholdet i listen etter regelen: Hver streng skal starte med stor bokstav, og skal ha stor bokstav like etter hvert mellomrom. Ellers skal strengene ha bare små bokstaver. Eksempel på kjøring av funksjonen fra konsollen i Spyder:

```
In[4]: gruppeledere = ['sAndRa LEkVe',
                       'maTHilde bloM bergenHEim']
In[5]: korrigjer(gruppeledere)
In[6]: gruppeledere
Out[6]: ['Sandra Lekve', 'Mathilde Blom Bergenheim']
```

2 Yatzy

Svar leveres på fil med navn `oppg2.py`

- a) (10%) Skriv en funksjon `kast` som tar en heltallsparameter `n`, og som re-turnerer et kast med `n` terninger i form av en liste med `n` tilfeldige heltallsverdier i intervallet 1–6. Eksempel på kjøring av funksjonen fra konsollen i Spyder:

```
In[9]: kast(5)
Out[9]: [2, 6, 3, 1, 6]
```

- b) (10%) Skriv en funksjon `finn_antall` som tar som parameter en liste (`terninger`) med terningkast av den typen funksjonen `kast` returnerer. Funksjonen skal returnere en heltallsliste (`antall`) av lengde 6. Første element i `antall`, dvs. `antall[0]`, skal være lik antall enere i `terninger`. Neste element i `antall`, dvs. `antall[1]`, skal være lik antall toere i `terninger`, osv., og siste element i `antall`, dvs. `antall[5]`, skal være lik antall seksere i `terninger`. Eksempel på kjøring av funksjonen fra konsollen i Spyder:

```
In[10]: finn_antall([2, 6, 3, 1, 6])
Out[10]: [1, 1, 1, 0, 0, 2]
```

- c) (10%) Skriv en funksjon `finn_flest` som tar som parameter en heltallsliste

(**antall**) av den typen funksjonen **finn_antall** returnerer. Husk at den listen inneholder antall enere, toere, ..., seksere. Funksjonen skal returnere hvilken terningverdi (1–6) som forekommer flest ganger. Eksempler på kjøring av funksjonen fra konsollen i Spyder:

```
In[11]: finn_flest([1, 1, 1, 0, 0, 2])
Out[11]: 6
In[12]: finn_flest([0, 1, 2, 0, 2, 0])
Out[12]: 5
```

Velg selv hvilken verdi **finn_flest** skal returnere når svaret er tvetydig (i siste eksempel kunne returverdien vært 3).

- d) (10%) Skriv en funksjon **nytt_kast** som tar som første parameter en liste (**terninger**) med terningkast av den typen funksjonen **kast** returnerer. Som andre parameter tar funksjonen en heltallsparameter (**spar**) med verdi i intervallet 1–6. Funksjonen gjør et nytt kast med alle terningene som ikke har samme verdi som **spar**, og oppdaterer verdien i **terninger** til det nye kastet. Eksempler på kjøring av funksjonen fra konsollen i Spyder:

```
In[12]: terninger = [2, 6, 3, 1, 6]
In[13]: nytt_kast(terninger, 6)
In[14]: terninger
Out[14]: [5, 6, 2, 3, 6]
In[15]: nytt_kast(terninger, 6)
In[16]: terninger
Out[16]: [1, 6, 5, 4, 6]
In[17]: nytt_kast(terninger, 6)
In[18]: terninger
Out[18]: [3, 6, 3, 3, 6]
In[19]: nytt_kast(terninger, 3)
In[20]: terninger
Out[20]: [3, 3, 3, 3, 4]
In[21]: nytt_kast(terninger, 3)
In[22]: terninger
Out[22]: [3, 3, 3, 3, 1]
```

- e) (10%) Skriv en funksjon **yatzy** som tar et antall (**n**) terninger som parameter, og som lar brukeren trille terningene like til hun får yatzy, dvs. alle **n** terningene viser samme verdi. I hver omgang skal funksjonen velge en verdi 1–6 som skal spares, slik at et maksimalt antall terninger som viser samme verdi legges til side. Funksjonen skal returnere antall omganger som minst en terning måtte kastes før yatzy ble oppnådd. I hver omgang skal funksjonen skrive ut hva terningene viser. Eksempel på kjøring av funksjonen fra konsollen i Spyder:

```
In[23]: yatzy(5)
[6, 3, 1, 2, 1]
[4, 6, 1, 4, 1]
[4, 3, 2, 4, 4]
[4, 5, 1, 4, 4]
[4, 3, 6, 4, 4]
[4, 5, 4, 4, 4]
[4, 4, 4, 4, 4]
Out[23]: 7
```

- f) (10%) Bruk funksjonen `yatzy` ved å kalle den i slutten av programfilen `oppg2.py`. Eksempel på dialog:

```
Antall terninger: 5
[6, 3, 1, 2, 1]
[4, 6, 1, 4, 1]
[4, 3, 2, 4, 4]
[4, 5, 1, 4, 4]
[4, 3, 6, 4, 4]
[4, 5, 4, 4, 4]
[4, 4, 4, 4, 4]
Yatzy med 5 terninger på 7 kast.
```

- g) (0%) *Frivillig ekstraoppgave for de som har lyst til å gå utenfor pensum. Temaet i ekstraoppgaven, plotting, er en del av pensum for kurset INF621 (Modul 2).* Lag et plott over antall omganger du trenger for å få yatzy med `n` terninger som funksjon av `n` på intervallet 1–100. For å plote verdiene i en liste `omganger`, bruk

```
from matplotlib import pyplot
pyplot.plot(omganger)
```

Fjern skjermdumpen i funksjonen `yatzy` når du løser ekstraoppgaven!