

# Øving 3

INF620 - Høstsemesteret 2021

Øvingsoppgavene er ikke obligatoriske, men vi anbefaler likevel at du gjør de og leverer de innen fristen — Den eneste måten å lære å programmere på er ved å programmere. Ved å gjøre oppgavene får du også testet deg selv og sjekket at du forstår begrepene. Du skal levere én zip-fil, `oving3.zip`, som inneholder de 4 filene `oppg1.py`–`oppg4.py`. For å komprimere en eller flere filer til en zip-fil høyreklikker du filene (i dette tilfellet `oppg1.py`–`oppg4.py`) i maskinens filnavigasjonsprogram og velger **Komprimer** eller **Send til → Komprimert mappe**. Frist: Torsdag 30. september kl 23:59

## 1 Sjekk av gyldig passord (30%)

Svar leveres på fil med navn `oppg1.py`

Vi sier at et passord er gyldig dersom

- det inneholder minst en liten bokstav,
- det inneholder minst en stor bokstav,
- det inneholder minst et siffer,
- det inneholder minst et tegn som verken er bokstav eller siffer,
- og det består av minst 8 karakterer.

Skriv et program som leser inn en streng, og som skriver til konsollen om strengen er et gyldig passord eller ikke.

**Hint:** Nyttige boolske metoder tilgjengelige for enhver strengvariabel `streng`: `streng.islower()`, `streng.isupper()` og `streng.isdigit()`.

Eksempler på virkemåte:

```
Passord: inf620
Ugyldig passord.
```

```
Passord: iNf6$!
Ugyldig passord.
```

Passord: iNfxx2\$!  
Gyldig passord.

## 2 Terningkast: Tre like verdier (30%)

Svar leveres på fil med navn `oppg2.py`

Vi skal undersøke hvor mange kast med tre terninger vi trenger før alle terningene viser samme verdi. Skriv et python-program som gjentatt genererer tre tilfeldige heltallige verdier i intervallet 1–6, like til alle tre verdiene er like. Antall gjentakelser skal skrives ut sammen med den felles verdien som ble generert i siste gjentakelse.

Eksempel på utskrift fra programmet:

```
1. kast: 2 4 3
2. kast: 3 1 5
3. kast: 4 1 1
4. kast: 1 4 2
5. kast: 1 3 5
6. kast: 5 1 2
7. kast: 4 1 6
8. kast: 6 6 2
9. kast: 4 2 3
10. kast: 2 2 2
Etter 10 kast med tre terninger viste alle terningene verdien 2.
```

```
1. kast: 2 5 4
2. kast: 6 4 4
3. kast: 5 5 5
Etter 3 kast med tre terninger viste alle terningene verdien 5.
```

## 3 Terningkast: Tre seksere (40%)

Svar leveres på fil med navn `oppg3.py`

Problemstillingen er den samme som i oppgave 2, bortsett fra at

- programmet stopper først når alle terningene viser verdien 6, og
- så snart en terning viser verdien 6, skal den ikke kastes på nytt.

I hver omgang skal alle terningenes verdier skrives ut.

Eksempel på utskrift fra programmet:

```
1. kast: 2 6 1
2. kast: 1 6 6
3. kast: 4 6 6
4. kast: 1 6 6
5. kast: 6 6 6
Etter 5 kast med tre terninger viste alle terningene 6.
```

```
1. kast: 2 4 1
2. kast: 3 1 6
3. kast: 2 4 6
4. kast: 3 2 6
5. kast: 5 3 6
6. kast: 3 1 6
7. kast: 4 2 6
8. kast: 2 3 6
9. kast: 2 4 6
10. kast: 2 4 6
11. kast: 6 4 6
12. kast: 6 6 6
Etter 12 kast med tre terninger viste alle terningene 6.
```

## 4 Sorteringsgrad av navn (0%)

*Dette er en ekstraoppgave for de som vil ha litt mer arbeid, og kan løses med det vi har lært så langt. Svar kan leveres på fil med navn **oppg4.py**.*

Vi sier at *sorteringsgraden* til et navn er antall par av nabobokstaver i navnet som står alfabetisk sorterte.

*Eksempel:*

1. 'Stein' har sorteringsgrad 3 (sorterte nabopar: 's'-'t', 'e'-'i', 'i'-'n')
2. 'Elise' har sorteringsgrad 2 (sorterte nabopar: 'e'-'l', 'i'-'s')
3. 'Erlend' har sorteringsgrad 2 (sorterte nabopar: 'e'-'r', 'e'-'n')
4. 'Mathilde' har sorteringsgrad 4 (sorterte nabopar: 'a'-'t', 'h'-'i', 'i'-'l', 'd'-'e')
5. 'Ida' har sorteringsgrad 0
6. 'Anne' har sorteringsgrad 2 (sorterte nabopar: 'a'-'n', 'n'-'n')

Skriv et program som leser inn navn fra tastaturet, og som skriver ut navnenes sorteringsgrad. Programmet skal avsluttes når et tomt navn leses inn. Eksempel:

Første navn: Mathilde  
Mathilde er et navn med sorteringsgrad 4.  
Neste navn: Fredrik  
Fredrik er et navn med sorteringsgrad 3.  
Neste navn: Dusan  
Dusan er et navn med sorteringsgrad 2.  
Neste navn: Mathias  
Mathias er et navn med sorteringsgrad 3.  
Neste navn: Sandra  
Sandra er et navn med sorteringsgrad 2.  
Neste navn: Ida  
Ida er et navn med sorteringsgrad 0.  
Neste navn:

Regler:

1. Om navnet er skrevet med små eller store bokstaver, eller en kombinasjon, skal ikke ha innvirkning på svaret.
2. Vi sier at dersom to nabobokstaver er like, så står de alfabetisk sorterte.
3. Det er tilstrekkelig at programmet gir korrekt svar for navn bestående av bokstavene 'a'-'z' og 'A'-'Z'.