

## INFO132 Laboppgaver 7

(Noen oppgaver kan være basert på AI-generert tekst, og kvalitetssikret av en lærer)

### Enkle oppgaver

1. Lag en funksjon som teller antall ord i en tekst

```
>>> antallOrd('Vått og kaldt og breiflabb overalt')
6
```

2. Lag en funksjon som skriver ut en tekst der ordene skrives vekselvis med store og små bokstaver

```
>>> skrivStoreSmå('Vått og kaldt og breiflabb overalt')
VÅTT og KALDT og BREIFLABB overalt
```

Hint: metodene `str.upper` og `str.lower`

3. Lag en funksjon som gitt et antall, et adjektiv og navnet på en plass returnerer en tekst som i disse eksemplene:

```
>>> musikanter(3, 'små', 'høybro')
'3 små musikanter på høybro plass'
>>>
>>> musikanter('ni', 'danske', 'danmarks')
'ni danske musikanter på danmarks plass'
```

Bruk f-strenger.

4.

a) Lag en funksjon `formater_navn` som tar inn fornavn og etternavn som strenger og returnerer en formatert streng på følgende måte: 'Etternavn, Fornavn'.

```
>>> formater_navn('Ole', 'Nordmann')
'Nordmann, Ole'
```

b) Utvid funksjonen fra a) slik at den også tar hensyn til mellomnavn hvis det er gitt, mellomnavnet skal bli gitt som et tredje argument.

```
>>> formater_navn('Ole', 'Nordmann', 'Johan')
'Nordmann, Ole Johan'
```

## Middels vanskelige oppgaver

5. Lag en funksjon som gitt et fullt navn vil skrive det ut med etternavnet først, deretter fornavnet, evt etterfulgt av forbokstaven i alle mellomnavn

```
>>> skrivNavn('Ole Martin Eek Olsen')
Olsen, Ole M. E.
```

6. Lag en funksjon som tar inn en liste av tekststrenger samt et heltall N som argumenter og returnerer en ny liste med de av tekststrengene som har en lengde på minst N tegn.

```
>>>liste
['Vått', 'og', 'kaldt', 'og', 'breiflabb', 'overalt']
>>>lengre_enn(liste,5)
['kaldt', 'breiflabb', 'overalt']
>>>lengre_enn(liste,7)
['breiflabb', 'overalt']
```

7. Lag en funksjon som skriver ut teksten til sangen 'Tre små musikanter' der man kan velge å erstatte vokalene i teksten med en oppgitt vokal som opsjon

```
>>> treSmåMusikanter()
tre små musikanter på høybro plass
sto og spilte på en kontrabass
så kom en konstabel, spurte hva var hendt
tre små musikanter på høybro plass
>>>
>>> treSmåMusikanter('i')
tri smi misikintir pi hiibri pliss
sti ig spilti pi in kintribiss
si kim in kinstibil, spirti hvi vir hindt
tri smi misikintir pi hiibri pliss
```

8. Lag en funksjon som, gitt en liste av kongelige på formen 'Tittel Navn', skriver ut navnet (uten tittel) til alle prinser og prinsesser.

Ta høyde for at titlene kan skrives med stor eller liten stor forbokstav.

```
>>>Kongefamilie
['Kong Otto', 'dronning Magda', 'Kronprins Ulf',
 'kronprinsesse Frida', 'prinsesse Angela', 'Prins Peter',
 'prinsesse Petra', 'grevinne Gunda', 'Grev Wilhelm',
 'Prins Ingolf']
>>>
>>> prinserOgPrinsesser(Kongefamilie)
Angela
Peter
Petra
```

Hint: metoden `str.startswith`

9. Lag en funksjon som gitt en tekst returnerer teksten reversert

```
>>>snu('Python')
'nohtyP'
```

10. Lag en funksjon som sjekker om et ord er et palindrom. Dvs at det leses likt forlengs og baklengs

```
>>> palindrom('pip')
True
>>> palindrom('regninger')
True
>>> palindrom('regning')
False
```

## Vanskelige oppgaver

11. Lag en funksjon som tar inn en tekststreng og sjekker hvor mange ganger hvert tegn forekommer i strengen. For hvert tegn skal funksjonen skrive ut antallet ganger det forekommer i teksten.

```
tell_tegn('programmering')
p : 1
r : 3
o : 1
g : 2
a : 1
m : 2
e : 1
i : 1
n : 1
```

12. Lag en funksjon `mest_brukte_ord` som tar en setning/streng og returnerer det mest brukte ordet i den setningen. Hvis det er flere ord som forekommer flest ganger er det vilkårlig hvilket som blir returnert.

```
>>>mest_brukte_ord('Dette er en test som er enkel og grei')
'er'
```

*Hint:* `split()` og `count()`

13. Lag en funksjon som gitt en tekststreng og et tall `N` returnerer listen med alle delstrenger (utsnitt) av lengde `N`.

```
>>>delstrenger('Vått og kaldt',4)
['Vått', 'ått ', 'tt o', 't og', ' og ', 'og k', 'g ka',
' kal', 'kald', 'aldt']
```

## Ekstraoppgaver

14.

*Tulleveien Velforening* registrerer hvem som holder ulike verv i en tekstvariabel kalt TV

```
TV=      \
'''
Tulleveien Velforening
leder: Kari
kasserer: Ole
IT-ansvarlig: Liv
parkeringsansvarlig: Kari
arrangementsansvarlig: Liv
hagekonsulent: Kari
brannansvarlig: Kari
'''
```

Husk at trippel anførselstegn

''' ... '''

brukes for tekster som kan  
strekke seg over flere linjer

a) (enkel)

Lag en funksjon som gitt en person returnerer antallet verv personen har.

```
>>> antallVerv('Kari')
4
```

b) (middels vanskelig)

Lag en funksjon som returnerer listen med personene som har verv

```
>>> personer()
['Kari', 'Ole', 'Liv']
```

c) (vanskelig)

Lag en funksjon som gitt en person og en liste av verv, innsetter personen i alle vervene i listen. Eksempel: Per innsettes som kasserer og hagekonsulent:

```
>>> TV=innsett('Per',['kasserer','hagekonsulent'])
>>> print(TV)
Tulleveien Velforening
leder: Kari
kasserer:Per
IT-ansvarlig: Liv
parkeringsansvarlig: Kari
arrangementsansvarlig: Liv
hagekonsulent:Per
brannansvarlig: Kari
```

15. (Middels vanskelig)

Lag en funksjon `flett(t1, t2)` som fletter tekstene `t1` og `t2` sammen til 1 tekst

Dvs den skal returnere en tekst som starter med det 1.tegnet i `t1` etterfulgt av 1. tegnet i `t2`. Så kommer 2.tegn i `t1`, så 2. tegn i `t2` og så videre. Dersom `t1` og `t2` ikke er like lange skal resten fra den lengste legges til slutt i resultatet

```
>>> flett('abcdefgh', '12345')
'a1b2c3d4e5fgh'
>>>
```

16. (Vanskelig) For de som ønsker en utfordring

*Cæsar skift* er en enkel form for kryptering av ord der hver bokstav erstattes av bokstaven som ligger N plasser lengre ut i alfabetet. Hvis det ikke er nok bokstaver lengre utover fortsetter man fra begynnelsen av alfabetet.

Nå skal du lage funksjoner for kryptering med  $N=1$ , altså at hver bokstav erstattes med den neste i alfabetet, og at 'å' erstattes med 'a'.

Du kan ta for gitt at input-ordet ikke inneholder tall, mellomrom eller andre tegn enn de norske bokstavene fra a-å.

a) Lag en funksjon som gitt en tekst returnerer den krypterte versjonen av teksten.

```
>>>krypter('programmering')
'qsphsbnnfsjoh'
```

Hint: hvis du vil prøve å ta høyde for både store og små bokstaver i ordet kan metoden `str.isupper()` være nyttig. (Sjekk `>>>help(str.isupper())` i IDLE)

b) Lag en funksjon som de-krypterer en tekst. Med andre ord: transformerer tilbake til klartekst. (Hint: Cæsar skift med  $N=-1$ )

```
>>>dekrypter('qsphsbnnfsjoh')
'programmering'
```

17. (Vanskelig) Nøtt for spesielt interesserte

Bytt tekstverdier i to variable uten å innføre en ekstra hjelpevariabel.

```
a='hei'
b='hopp'
print(a,b)
a= ... fyll inn det som mangler ...
b= ... fyll inn det som mangler ...
a= ... fyll inn det som mangler ...
print(a,b)
```

Fullfør programmet slik at det gir denne utskriften

```
hei hopp
hopp hei
```