

INFO132 oppgavesett om fortegnelser

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

Enkle oppgaver

1.

a) Lag en fortegnelse over hvor mange dager det er i hver måned (se bort fra skuddår)

b) Skriv en funksjon som lar en bruker taste inn nedbørsmengden for et gitt antall dager og lager en fortegnelse over nedbørsdataene for perioden. Eksempel:

```
>>> antallDager['mai'] # a)
31
>>>
>>> treSisteDager=lagNedbørsFortegnelse(3) # b)
dag 1, nedbør: 5
dag 2, nedbør: 2
dag 3, nedbør: 3
>>> treSisteDager
{1: 5, 2: 2, 3: 3}
>>>
>>>
>>> nedbørMai=lagNedbørsFortegnelse(antallDager['mai'])
dag 1, nedbør: 2
dag 2, nedbør: 5
dag 3, nedbør: 6
    ⋮
dag 30, nedbør: 2
dag 31, nedbør: 1
>>>
>>> nedbørMai
{1: 2, 2: 5, 3: 6, ..., 30: 2, 31: 1}
```

2. Du har fortegnelser med nedbørsdata for hver dag i en måned. For eksempel:

```
nedbørMai={1:2, 2:5, 3:6, 4:4, 5:2, 6:0,7:0, 8:1,9:2,10:9,
           11:5, 12:5, 13:0, 14:0, 15:3, 16:3,17:0, 18:4,
           19:5, 20:9, 21:1, 22:4, 23:0, 24:0, 25:0, 26:2,
           27:5, 28:4, 29:0, 30:2, 31:1}
```

Lag en funksjon som beregner gjennomsnittlig nedbør i perioden

```
>>> snittNedbør(nedbørMai)
2.71
```

3. Lag en funksjon som tar inn en fortegnelse med navn på personer og deres alder. Den skal returnere en liste med navnene på personene som er over en viss alder. Alderen som skal sjekkes skal også være en parameter for funksjonen.

```
>>> alderfortegnelse
{'Jonas': 25, 'Lise': 30, 'Per': 20, 'Mona': 35}
>>>
>>> person_over_alder(alderfortegnelse, 30)
['Lise', 'Mona']
```

4. Lag en funksjon som tar inn en tekststreng og returnerer en fortegnelse som inneholder hvert unike ord i teksten som nøkler og antall forekomster av hvert ord som verdi.

```
>>> tell_forekomster('Vått og kaldt og breiflabb overalt')
{'Vått': 1, 'og': 2, 'kaldt': 1, 'breiflabb': 1, 'overalt': 1}
```

5. Tenk deg en konkurranse der deltakerne oppnår poengsummer i hver runde. Lag en funksjon `poengSnitt(deltakere)` som tar inn en fortegnelse med deltakere. Deltakerens navn er nøkkel for en liste av resultatverdier. Eksempel:

```
deltakere={'Liv':[5,9,5], 'Ola':[4,3,8], 'Gro':[2,6,9], 'Jon':[5,6,4]}
```

Funksjonen skal returnere en fortegnelse der nøklene er deltakernavn og verdien er gjennomsnittet av deltakerens poengsummer.

```
>>> poengSnitt(deltakere)
{'Liv': 6.33, 'Ola': 5.0, 'Gro': 5.67, 'Jon': 5.0}
```

Middels vanskelige oppgaver

6.

Lag en funksjon `FibonacciFortegnelse(n)` som returnerer fortegnelsen med de n første Fibonacci-tallene. (Husk: Fibonacci-serien starter slik: 1, 1, 2, 3, 5, 8.... og generelt er hvert tall summen av de to foregående). Eksempel:

```
>>> FibonacciFortegnelse(5)
{0: 1, 1: 1, 2: 2, 3: 3, 4: 5, 5: 8}
>>>
>>> # skriver de 8 første Fibonacci-tallene
>>> for x in FibonacciFortegnelse(8).values():print(x)
1
1
2
3
5
8
13
21
-
```

7. Du har to fortegnelser A og B der verdiene i A er av samme type som nøklene i B. For eksempel:

```
dager={1:'man',2:'tir', 3:'ons', 4:'tor',
        5:'fre',6:'lør', 7:'søn'}

temperaturer={'man':11, 'tir':9, 'ons':7, 'tor':12,
              'fre':11, 'lør':9, 'søn':8}
```

Da kan disse dataene *kobles* til en ny fortegnelse C, der nøkler fra A er assosiert med verdier fra B. Spesifikt: hvis $x:y$ er i A og $y:z$ er i B så skal $x:z$ være i C. Med andre ord, hver gang en verdi i A matcher en nøkkel i B så skal C assosiere A-nøkkelen med B-verdien.

Lag en funksjon for å koble to fortegnelser. Eksempler:

```
>>> koble(dager,temperaturer)
{1: 11, 2: 9, 3: 7, 4: 12, 5: 11, 6: 9, 7: 8}
>>>
>>> koble(temperaturer,dager)
{'ons': 'søn'}
```

8. I et sosialt nettverk kan en person følge andre personer. Dette er ikke nødvendigvis gjensidig. En følge-fortegnelse har personers navn som nøkler knyttet til lister av navn på personene de følger.

Eksempel

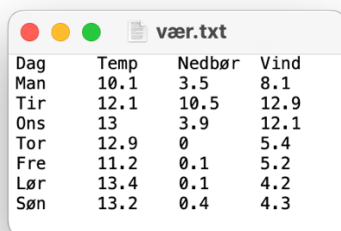
```
følgere = {'Anne': ['Berit', 'Karl', 'Dag'], 'Berit': ['Karl', 'Ina'],
           'Karl': ['Anne'], 'Dag': ['Anne', 'Ina', 'Berit'],
           'Ina': ['Karl', 'Berit', 'Dag']}
```

Dvs: Anne følger Berit, Karl og Dag, men bare Karl og Dag følger Anne gjensidig, osv.

Lag en funksjon `gjensidige_følgere(følgefortegnelse, navn)` som gitt en følge-fortegnelse og en person skal returnere listen med personer som følger hen gjensidig.

```
>>>gjensidige_følgere(følgere, 'Anne')
['Karl', 'Dag']
>>>gjensidige_følgere(følgere, 'Berit')
['Ina']
>>>gjensidige_følgere(følgere, 'Jens')
'ukjent person: Jens'
```

9. Filen `vær.txt` inneholder vær-data for en uke



Dag	Temp	Nedbør	Vind
Man	10.1	3.5	8.1
Tir	12.1	10.5	12.9
Ons	13	3.9	12.1
Tor	12.9	0	5.4
Fre	11.2	0.1	5.2
Lør	13.4	0.1	4.2
Søn	13.2	0.4	4.3

a) Lag en funksjon som leser en slik fil og returnerer en nestet fortegnelse som dette

```
>>>vær=lesVær('vær.txt')
>>>vær
{'Man': {'Temp': 10.1, 'Nedbør': 3.5, 'Vind': 8.1},
 'Tir': {'Temp': 12.1, 'Nedbør': 10.5, 'Vind': 12.9},
 'Ons': {'Temp': 13.0, 'Nedbør': 3.9, 'Vind': 12.1},
 'Tor': {'Temp': 12.9, 'Nedbør': 0.0, 'Vind': 5.4},
 'Fre': {'Temp': 11.2, 'Nedbør': 0.1, 'Vind': 5.2},
 'Lør': {'Temp': 13.4, 'Nedbør': 0.1, 'Vind': 4.2},
 'Søn': {'Temp': 13.2, 'Nedbør': 0.4, 'Vind': 4.3}}
```

b) Bruk fortegnelsen til å finne temperaturen på tirsdag og vindstyrken på fredag.

c) Skriv ut dagene der nedbørsmengden er mindre enn 0.3 mm

Vanskelige oppgaver

10. Tenk deg nå en nestet fortegnelse der ulike sosiale nettverk er knyttet til sine respektive følge-
fortegnelser som beskrevet i oppgave 8. Eksempel

```
nettverk=\n{'Facebook':{'Anne':['Berit','Karl','Dag'],'Berit':['Karl','Ina'],'Karl':['Anne'],\n            'Dag':['Anne','Ina','Berit'],'Ina':['Karl','Berit','Dag']},\n\n  'Instagram':{'Anne':['Ina','Eva','Dag'],'Berit':['Dag'],'Gro':['Ina'],\n              'Eva':['Anne','Dag'],'Dag':['Anne','Ina','Berit'],\n              'Ina':['Anne','Berit','Dag']},\n\n  'Threads':{'Jon':['Berit','Karl','Dag'],'Berit':['Karl','Ina'],\n            'Åse':['Dag','Jon'],'Karl':['Anne'],'Dag':['Jon','Ina','Berit'],\n            'Ina':['Karl','Berit','Dag']}} }
```

Da kan vi for eksempel bruke funksjonen fra oppgave 8 til å finne personers gjensidige følgere i ulike nettverk slik:

```
>>>gjensidige_følgere(netttverk['Facebook'],'Anne')\n['Karl','Dag']\n>>>gjensidige_følgere(netttverk['Instagram'],'Anne')\n['Ina','Eva','Dag']
```

a) Lag en funksjon `følgere` (*nettverksfortegnelse*, *sosialt_netttverk*, *person*) som returnerer listen med de som personen følger i det gitte sosiale nettverket. Ta høyde for at ikke alle personer deltar i alle sosiale nettverk. Ta høyde for at ikke alle personer deltar i alle sosiale nettverk.

```
>>>følgere(netttverk,'Facebook','Dag')\n['Anne','Ina','Berit']\n>>>følgere(netttverk,'Threads','Dag')\n['Jon','Ina','Berit']\n>>>følgere(netttverk,'Instagram','Jens')\nJens er ikke på Instagram\n[]\n>>>følgere(netttverk,'Snapchat','Anne')\nukjent nettverk: Snapchat\n[]
```

b) Lag en funksjon `alle_fulgte` (*nettverksfortegnelse*, *person*) som skriver ut navn på alle sosiale nettverk som personen er på, samt navn på alle hen følger på minst ett av nettverkene. Unngå duplikater.

```
>>>alle_fulgte(netttverk,'Anne')\nAnne er på disse nettverkene: Facebook, Instagram\nog følger disse personene: Berit, Karl, Dag, Ina, Eva\n>>>\n>>>alle_fulgte(netttverk,'Jon')\nJon er på disse nettverkene: Threads\nog følger disse personene: Berit, Karl, Dag
```

Ekstra oppgaver

11. (enkel)

a) Lag en funksjon `listeFortegnelse(Liste)` som oversetter listen til en tilsvarende fortegnelse. Det vil si at nøklene i fortegnelsen skal svare til indeksene i listen.

b) Generelt har ikke elementene i en fortegnelse en innbyrdes rekkefølge, slik som elementene i en liste har. Derfor kan ikke begrepet utsnitt (slice) overføres fra lister, der et utsnitt defineres ved en segment-indeksering [*fra-indeks:til-indeks*]. Men, når nøklene i fortegnelsen tilsvarer indeksene i en liste, som i punkt a), kan vi tenke oss tilsvarende utsnitt i fortegnelsen.

Lag en funksjon `utsnitt(listefortegnelse, fra, til)` slik at

`utsnitt(listeFortegnelse(Liste), fra, til)` tilsvarer `Liste[fra:til]`

Eksempler:

```
>>> # a)
>>> L=['Per','Liv','Kari','Ola']
>>> F=listeFortegnelse(L)
>>> F
{0: 'Per', 1: 'Liv', 2: 'Kari', 3: 'Ola'}
>>> L[2]
'Kari'
>>> F[2]
'Kari'
>>>
>>>>> for i in range(0,len(L)):print(L[i])
Per
Liv
Kari
Ola
>>> for i in range(0,len(F)):print(F[i])
Per
Liv
Kari
Ola
>>>
>>> # b)
>>> L[1:3]
['Liv', 'Kari']
>>> utsnitt(F,1,3)
{1: 'Liv', 2: 'Kari'}
```

12. (middels vanskelig)

a) Lag en funksjon som skriver ut en fortegnelse sortert med hensyn på nøkkelverdien.

I resten av oppgaven ser vi på ‘en-til-en’-fortegnelser, der hver verdi forekommer kun 1 gang.

Eksempel:

```
engelskeSiffer={3:'three', 4:'four', 5:'five',
                0:'zero',1:'one', 2:'two',
                6:'six', 7:'seven',8:'eight',9:'nine'}
```

b) Lag en funksjon som gitt en en-til-en-fortegnelse, oppretter den *inverse* fortegnelsen der nøklene har blitt verdier, og omvendt.

c) Anta nå at du kun har den inverse fortegnelsen, og ikke originalen. Skriv ut den inverse fortegnelsen sortert med hensyn på verdien, dvs skriv ut `zero 0` først osv.

Eksempler

```
>>> # a)
>>> skrivSortert(engelskeSiffer)
0 zero
1 one
2 two
3 three
4 four
5 five
6 six
7 seven
8 eight
9 nine
>>>
>>>
>>> # b)
>>> i=invers(engelskeSiffer)
>>> i
{'three': 3, 'four': 4, 'five': 5, 'zero': 0, 'one': 1, 'two': 2,
'six': 6, 'seven': 7, 'eight': 8, 'nine': 9}
>>>
>>> # c)
>>> del engelskeSiffer # originalen har gått tapt
>>>
>>> skrivInversSortert(i)
zero 0
one 1
two 2
three 3
four 4
five 5
six 6
seven 7
eight 8
nine 9
>>>
```