

### OBS!

En tekstfil er opprettet/lagret i et bestemt *tegnsett*. Det finnes mange ulike tegnsett og de har varierende utvalg av spesialtegn. For eksempel vil et norsk tegnsett ha bokstavene æ, ø og å, mens et engelsk tegnsett vil ha erstattet disse med andre spesialtegn. Ulike editorer, selv ulike versjoner av IDLE, kan være satt opp med ulike tegnsett, og dette kan gi opphav til problemer. For eksempel hvis programmet ditt søker mht norske tegn i en fil som er laget utfra et engelsk tegnsett. slike problemer unngås hvis du bruker test-data som er laget med samme editor som du skriver programmene dine i. Hvis du vet hvilket tegnsett filen er lagret i, for eksempel utf-8, kan du spesifisere tegnsettet vha opsjonen `encoding` i `open`. Eksempel:

```
fil=open(filnavn, encoding='utf-8')
```

Det er ingen garanti for at din IDLE er satt opp med samme tegnsett som test-dataene som er lagt ut på MittUib. I verste fall må du lage dine egne test-data, men heldigvis er det ingen av oppgavene her som krever store test-data-sett.

1. Last ned ordlistefilen 'ordliste.txt' fra kursets hjemmeside på MittUib og lagre den der du lagrer python-programmene dine. Lag et program som leser inn ordlisten og lar brukeren taste inn norske søkeord og svarer med den engelske oversettelsen. Prøv å finne alle oppslag i ordlisten som starter på brukerens søkeord.

```
Oppgi søkeord (avslutt med 'slutt')
søkeord: utsnitt
utsnitt = slice

søkeord: inn
inndata = input
inn-enhet = input device

søkeord: juba
juba finnes ikke i ordlisten

søkeord: t
tekstsegment = slice
tekstverdi = string
tilordning = assignment
tolk = interpreter
tom funksjon = void function
tom streng = empty string

søkeord: slutt
>>>
```

2.

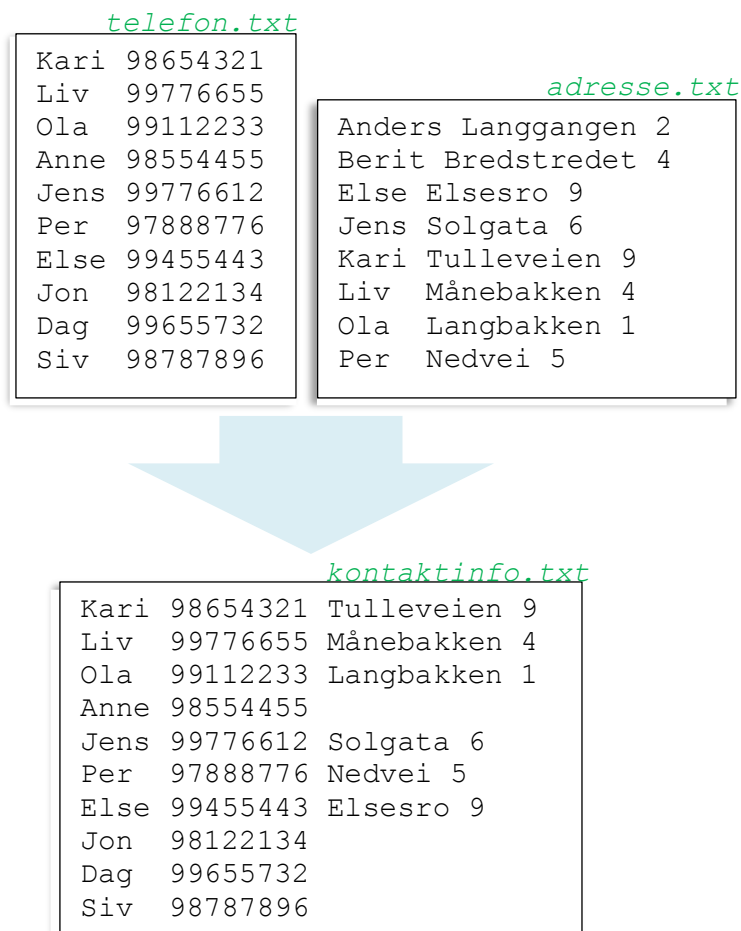
Vi har følgende to filer:

- telefon.txt der personer er registrert med navn og telefonnummer, i tilfeldig rekkefølge.
- adresse.txt der personer er registrert med navn og adresse. Denne filen er sortert alfabetisk

Adresse- og telefonlistene inneholder ikke nødvendigvis de samme personene.

Lag et program som tar utgangspunkt i personene i telefonlisten og lager en ny fil med kontaktinformasjon for disse. Alle skal være registrert med telefon samt adresse der den er kjent

Eksempel:



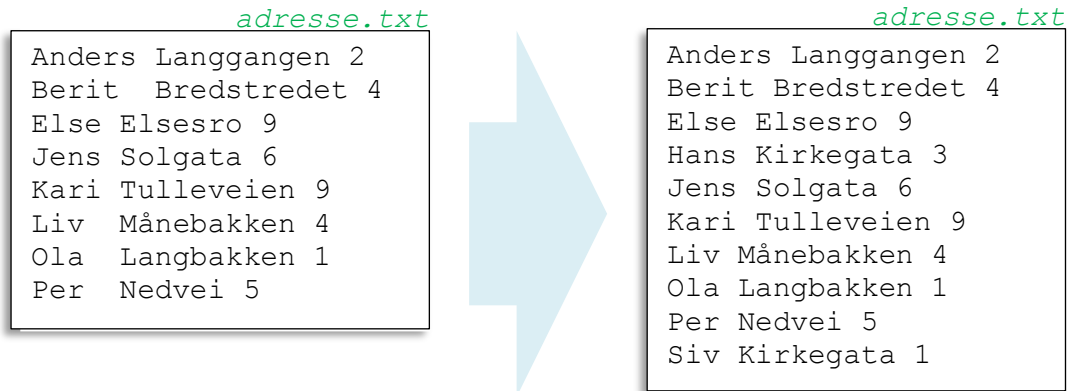
Adresselisten er sortert. Kan du utnytte det til å effektivisere programmet? Det vil si, unngå å utføre unødvendige operasjoner.

3.

a) Vi ser videre på den sorterte adresselisten `adresse.txt`

Lag en funksjon `settinn(person, fil)` som kan sette inn en ny person/adresse, på riktig sted i filen så den forblir sortert.

```
>>> settinn('Hans Kirkegata 3','adresse.txt')
>>> settinn('Siv Kirkegata 1', 'adresse.txt')
>>>
```

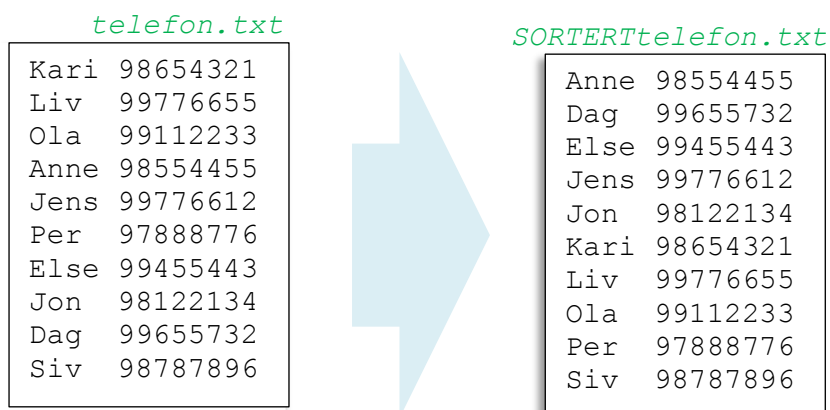


Funksjonen skal være generell nok til også å kunne virke med alle slags tekster på formen bokstaver-mellomrom-tegn, for eksempel `'Kari 98654321'`

b)

Bruk `settinn()` fra punkt a) til å lage en funksjon som kan lage en sortert versjon av `telefon.txt` på en ny fil `SORTERTtelefon.txt`.

```
>>> sorter('telefon.txt')
>>>
```



4.

Lag en funksjon som erstatter alle sifre i en fil med tilsvarende tallord, resultatet legges i en ny fil.

```
>>> erstattSifre('telefon.txt')  
>>>
```

*telefon.txt*

Kari	98654321
Liv	99776655
Ola	99112233
Anne	98554455
Jens	99776612
Per	97888776
Else	99455443
Jon	98122134
Dag	99655732
Siv	98787896



*NYTtelefon.txt*

Kari	ni	åtte	seks	fem	fire	tre	to	en
Liv	ni	ni	sju	sju	seks	seks	fem	fem
Ola	ni	ni	en	en	to	to	tre	tre
Anne	ni	åtte	fem	fem	fire	fire	fem	fem
Jens	ni	ni	sju	sju	seks	seks	en	to
Per	ni	sju	åtte	åtte	åtte	sju	sju	seks
Else	ni	ni	fire	fem	fem	fire	fire	tre
Jon	ni	åtte	en	to	to	en	tre	fire
Dag	ni	ni	seks	fem	fem	sju	tre	to
Siv	ni	åtte	sju	åtte	sju	åtte	ni	seks

5. Ta utgangspunkt i konto-eksemplet fra forelesningen. Lag et program for å administrere en ny bank sine kontoer. Her er eksempel på hvordan dialogen kunne se ut på bankens 1. dag.

```
>>> start()
-----
Velg handling (? for valgmeny)
1 - Opprett ny konto
2 - Innskudd
3 - Uttak
4 - Kontoutskrift
5 - Kontooversikt for kunde
6 - Avslutt
-----
Velg> 1
    Innehaver: Ole
    Startinnskudd: 1000
Velg> 1
    Innehaver: Kari
    Startinnskudd: 2000
Velg> 1
    Innehaver: Jens
    Startinnskudd: 3000
Velg> 5
    Kunde:          (kommentar: input <enter> gir alle kundene)
    1/Ole: 1000      (kommentar: kontonumre genereres her automatisk fra 1 og oppover)
    2/Kari: 2000
    3/Jens: 3000
Velg> 2
    Kontonummer: 2
    Beløp: 200
Velg> 3
    Kontonummer: 1
    Beløp: 100
Velg> 4
    Kontonummer: 1
    opprettet, saldo=1000
    1000-100
    saldo: 900
Velg> 1
    Innehaver: Guro
    Startinnskudd: 5000
Velg> 5
    Kunde: G         (kommentar: finner alle som begynner på G)
    4/Guro: 5000
Velg> 6
    takk for nå
```

Ved dagens slutt blir alle konto-opplysninger lagret på filen `kontoer.txt`. Det finnes også en fil `transaksjoner.txt` som oppdateres hver gang en handling utføres på en konto. Da legges kontonummer og handling til på slutten av filen. Filen brukes til å generere kontoutskrifter. Etter første dag så filene slik ut:

*kontoer.txt*

```
1
Ole
900
2
Kari
2200
3
Jens
3000
4
Guro
5000
```

*transaksjoner.txt*

```
1 opprettet, saldo=1000
2 opprettet, saldo=2000
3 opprettet, saldo=3000
2 2000+200
1 1000-100
4 opprettet, saldo=5000
```

Neste dag starter med at filene leses inn. Slik forløp den andre dagen:

```
>>> start()
-----
Velg handling (? for valgmeny)
1 - Opprett ny konto
2 - Innskudd
3 - Uttak
4 - Kontoutskrift
5 - Kontooversikt for kunde
6 - Avslutt
-----
Velg> 5
Kunde:
1/Ole: 900
2/Kari: 2200
3/Jens: 3000
4/Guro: 5000
Velg> 2
Kontonummer: 3
Beløp: 300
Velg> 3
Kontonummer: 1
Beløp: 1200
Ikke dekning
Velg> 3
Kontonummer: 1
Beløp: 120
Velg> 1
Innehaver: Britt
Startinnskudd: 2300
Velg> 5
Kunde:
1/Ole: 780
2/Kari: 2200
3/Jens: 3300
4/Guro: 5000
5/Britt: 2300
Velg> 4
Kontonummer: 1
opprettet, saldo=1000
1000-100
900-1200 avvist
900-120
saldo: 780
Velg> 6
takk for nå
```

Nå ser filene slik ut:

*kontoer.txt*

```
1
Ole
780
2
Kari
2200
3
Jens
3300
4
Guro
5000
5
Britt
2300
```

*transaksjoner.txt*

```
1 opprettet, saldo=1000
2 opprettet, saldo=2000
3 opprettet, saldo=3000
2 2000+200
1 1000-100
4 opprettet, saldo=5000
3 3000+300
1 900-1200 avvist
1 900-120
5 opprettet, saldo=2300
```