

1(a)

```

x = 2
y = '2'
a = [-2, x, y]

```

Anta at kodesnutten over har blitt køyrd, og at ei av setningane under er den neste setninga som blir utført. Kva blir skrive ut? Dersom programmet krasjar, skriv berre Error.

(Hugs at apostrofane og hermeteikna som omgir strengar i kildekoden *ikkje* blir inkludert i utskrifta.)

print(x)	
print(y)	
print('a')	
print(a[1])	
print(y + y)	
print(a[1] + x)	
print(a[a[1]] * 3)	

Maks poeng: 7

1(b)

```
a = 2
b = 2
a = a + b
b = a + b
a += 1
print(b - a)
```

Kva skriv dette programmet ut? (Dersom programmet krasjar, skriv berre Error)

Maks poeng: 2

1(c)

```
def incremented(x):  
    x += 1  
    return x  
def foo(x):  
    x = incremented(x)  
    y = incremented(x)  
    return x + y  
x = 3  
x = foo(x)  
print(x)
```

Kva skriv dette programmet ut? (Dersom programmet krasjar, skriv berre Error)

Maks poeng: 2

1(d)

```
a = ['a', 'b', 'c', 'd', 'e', 'f']
n = len(a) // 2
for i in range(n):
    x = a[i]
    a[i] = a[i*2]
    a[i*2] = x
r = ''
for c in a:
    r += c
print(r)
```

Kva skriv dette programmet ut? (Dersom programmet krasjar, skriv berre Error)

Maks poeng: 3

- 1(e)** Skriv ein funksjon **is_overweight** med to parameterar **height** og **weight**. Du kan anta at «height» og «weight» viser til talverdiar for respektive ein person sin høgde (i meter) og tyngde (i kilogram). Funksjonen skal returnere *True* dersom personen er overvektig, og *False* om ikkje.

Ein person er overvektig dersom kroppsmasseindeksen (KMI) til personen er 25 eller høgare. KMI er definert som **$KMI = \text{vekt}(\text{kg}) / \text{høgde}(\text{m})^2$**

For å få utteljing på denne oppgåva må alle testane (inkludert hemmelege testar) passere.

Maks poeng: 4

- 1(f)** Skriv ein funksjon **starts_with_h** med ein parameter **animals**. Du kan anta at *animals* peikar til ei liste med strengar. Funksjonen skal returnere ei ny liste som inneheld alle dyra som startar på bokstaven **'h'**.

For å få utteljing på denne oppgåva må alle testane (inkludert hemmelege testar) passere.

Maks poeng: 6

2(a)

```
# Hensikten er å be brukeren om to tall og skrive ut summen av tallene
int(x) = input('Det første tallet: ')
int(y) = input('Det andre tallet: ')
total = x + y
print(f'Summen er {total}')
```

Når programmet over blir kjørt, krasjar det med feilmeldinga:

```
File "/path/to/foo.py", line 2
  int(x) = input('Det første tallet: ')
  ^^^^^^^
SyntaxError: cannot assign to function call here. Maybe you meant '==' instead of '='?
```

Deloppgåve i:

- Forklar i korthet kva som er feil og korleis programmet kan rettast.

Avhengig av kva tankefeil vi har gjort, er det ikkje alltid forklaringsteksten i feilmeldinga og løysingsidéen den gir oss spesielt hjelpsam. Feilmeldinga seier likevel *noko* om kva som har gått gale.

Deloppgåve ii:

- Forklar kvifor det står «cannot assign to function call» i feilmeldinga.

Maksimalt 300 ord.

Skriv svaret ditt her

Maks poeng: 10

2(b) Anta at du har ei fil *foo.csv* med innhaldet under.

name,disipline,value,unit
Kari,long jump,5,m
Per,long jump,500,cm
Pål,long jump,5000,mm
Espen,standing long jump,123,cm

Du køyrer programmet som blir vist i vedlagde PDF. Kva blir innhaldet i *bar.csv*?

Skriv svaret ditt her

Maks poeng: 4

- 2(c)** Programmet i vedlagde PDF har dårlige variabelnamn, som gjør det vanskeleg å lese og forstå. Forklar rolla til dei ulike variablane i koden, og foreslå nye, sjølvbeskrivande namn for variabel- og funksjonsnamn.

PS: svar gjerne med ein tabell (kopier han inn i svaret ditt og fyll han ut):

Org. namn	Nytt namn	Forklaring av rolla til variablen
a		
b		
c		
d		
e		
f		
g		
h		

Skriv svaret ditt her

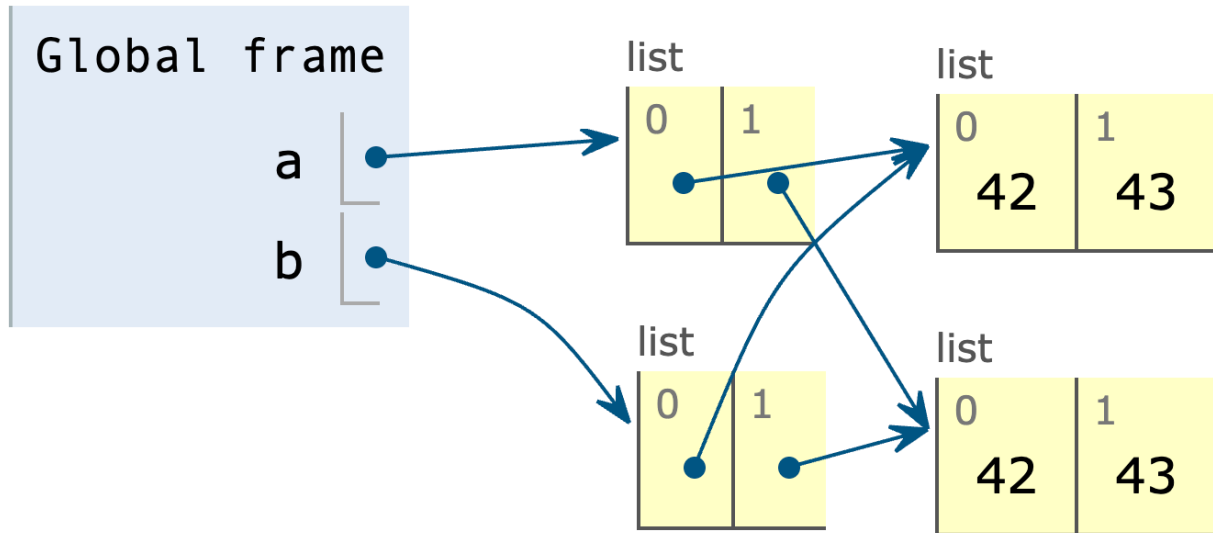
Maks poeng: 10

- 2(d)** Forklar hensikta med linje 17 i vedlagde PDF, og kom med eit forslag til ein betre streng du kan bruke. «I love crashing!» er trass alt ikkje særleg nyttig informasjon.

Skriv svaret ditt her

Maks poeng: 4

3(a)



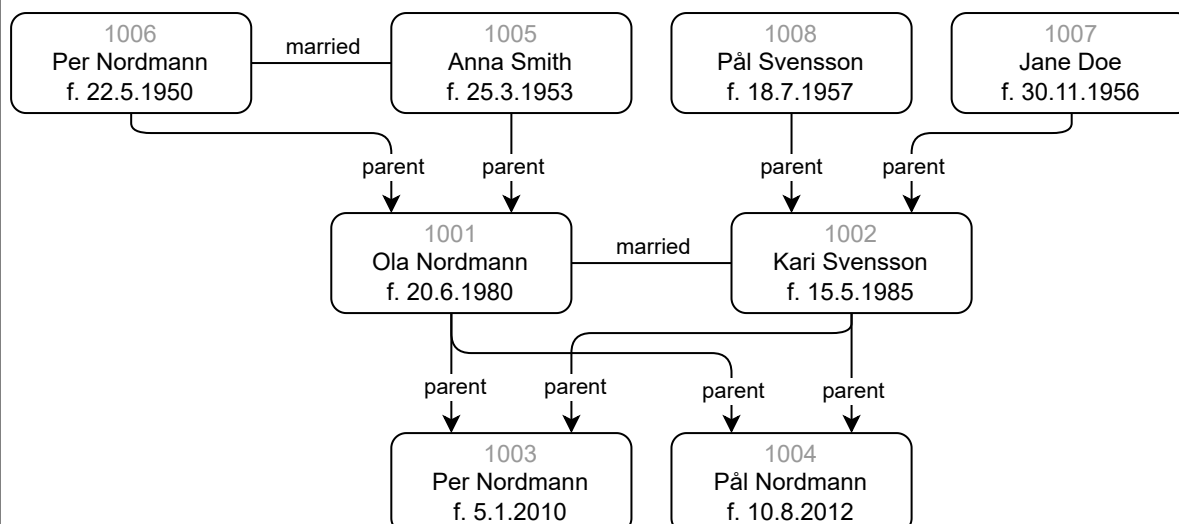
Skriv ein kodesnutt slik at minnet si tilstand blir som vist ovanfor. Du vil ikkje få trekk i poeng dersom du definerer andre variablar i tillegg.

Skriv svaret ditt her

Maks poeng: 3

3(b)

Ola og Kari har to søner, Per og Pål, som er oppkalla etter besteforeldra sine. Eit lite utsnitt av familietreet deira ser du her:



I fila *family.json* vedlagt ser du ein tekstuell representasjon av den same informasjonen som på biletet. Kvar person har ein unik id. Id-nummeret har derimot ingen faktisk betydning utover å bli brukt for intern organisering av fila. For enkelheits skuld kan du anta at alle foreldreskap som er beskrive i fila, er biologiske; altså at adoptivforeldre og liknande relasjonar ikkje er omtalte.

Oppgåvetekst

Ola har ein tvilling, Nora Nordmann. Korleis er det naturleg å utvide json-fila slik at Nora òg blir med i familietreet?

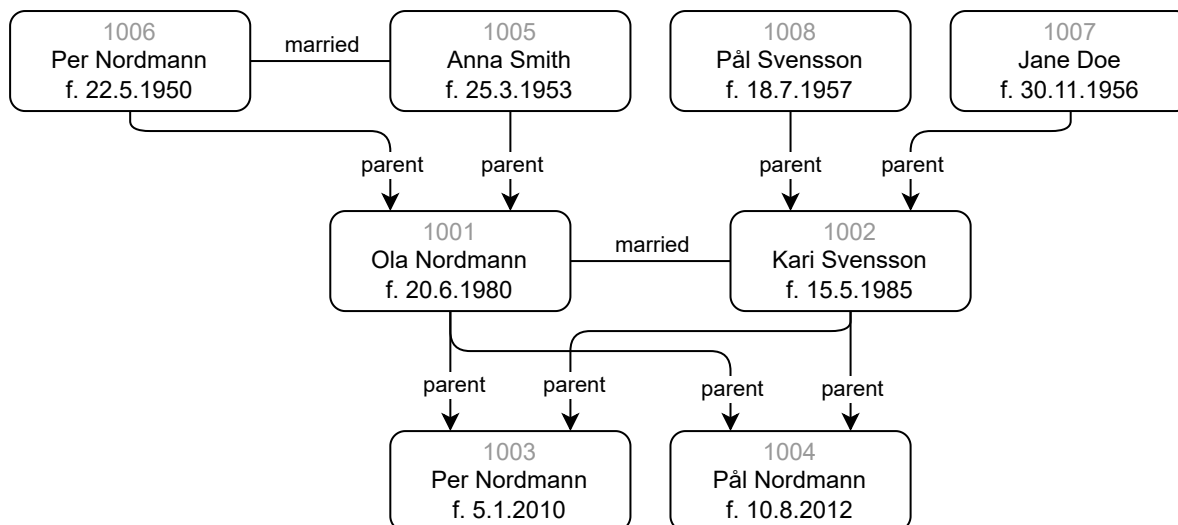
Du skal svare med å vise korleis json-fila ser ut etter endringane dine. Du treng ikkje å kopiere heile json-fila, berre vise korleis den nye informasjonen er skrive og vise/forklare kvar den befinn seg i forhold til resten av innhaldet i json-fila.

Skriv svaret ditt her

Maks poeng: 4

3(c)

Ola og Kari har to søner, Per og Pål, som er oppkalla etter besteforeldra sine. Eit lite utsnitt av familietreet deira ser du her:



I fila *family.json* vedlagt ser du ein tekstuell representasjon av den same informasjonen som på biletet. Kvar person har ein unik id. Id-nummeret har derimot ingen faktisk betydning utover å bli brukt for intern organisering av fila. For enkelheits skuld kan du anta at alle foreldreskap som er beskrive i fila, er biologiske; altså at adoptivforeldre og liknande relasjonar ikkje er omtalte.

Ola og Kari har blitt heilt rivne med, og legg til fleire og fleire familiemedlemmer. Til slutt har *family.json* blitt så stor at ho inneheld informasjon om tusenvis av personar, samt foreldreskap og ekteskap mellom dei.

Undervegs i arbeidet legg dei merke til at fleire personar har akkurat same namn; for eksempel heiter Per Nordmann (fødd 2010) nøyaktig det same som bestefaren sin (fødd 1950).

Oppgåvetekst

Skriv ein funksjon *get_ids* med to parameter:

- *tree*, eit oppslagsverk (dict) som representerer eit familietre. Du kan anta at det har same format som du ser eit eksempel på i *family.json*.
- *name*, ein streng (str) som representerar eit navn.

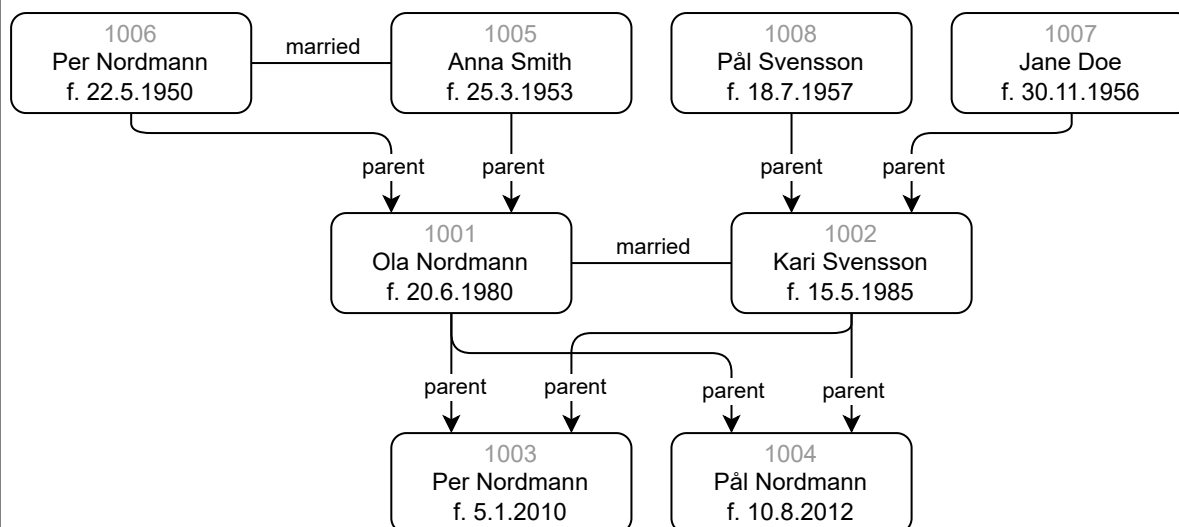
Funksjonen skal returnere ei liste med id-ar til personar med det gitte namnet. Til dømes: eit kall til *get_ids(tree, 'Per Nordmann')* skal returnere ei liste [1003, 1006].

Skriv svaret ditt her

Maks poeng: 7

3(d)

Ola og Kari har to søner, Per og Pål, som er oppkalla etter besteforeldra sine. Eit lite utsnitt av familietreet deira ser du her:



I fila *family.json* vedlagt ser du ein tekstuell representasjon av den same informasjonen som på biletet. Kvar person har ein unik id. Id-nummeret har derimot ingen faktisk betydning utover å bli brukt for intern organisering av fila. For enkelheits skuld kan du anta at alle foreldreskap som er beskrive i fila, er biologiske; altså at adoptivforeldre og liknande relasjonar ikkje er omtalte.

Ola og Kari har blitt heilt rivne med, og legg til fleire og fleire familiemedlemmer. Til slutt har *family.json* blitt så stor at ho inneheld informasjon om tusenvis av personar, samt foreldreskap og ekteskap mellom dei.

Oppgåvetekst

Ola og Kari har ein hypotese om at barn av ugifte foreldre (såkalla «kjærleiksbarn») har større sannsyn for å bli fødd om våren (i mars, april eller mai). Skriv eit program som finn

- kor mange kjærleiksbarn vi veit om totalt i slektstreet, og
- kor mange av dei som blei fødte om våren.

Vi definerer eit kjærleiksbarn som ein person der begge foreldra til personen er med i familietreet, men foreldra ikkje er markerte som gifte med kvarandre.

Du kan anta at du har tilgang på ei fil *family.json* med same format som vist i eksempelet. I eksempelet finn vi eitt kjærleiksbarn (Kari), og ho blei fødd om våren (i mai). Koden du skriv må likevel verke også når *family.json* beskriv eit slektstre med tusenvis av personar.

Du treng ikkje å ha optimal kjøretid for å få full uttelling på oppgåva. Det viktigaste er at løysinga er korrekt og lett å forstå. Sensor vil leggje vekt på at du klarer å dele opp oppgåva i eigna hjelpefunksjonar.

Skriv svaret ditt her

Maks poeng: 14

- 4 I denne oppgåva blir det lagt inn poeng frå arbeidet ditt med labar du har gjort gjennom semesteret.



CC BY-NC 4.0 kot-spit-klaviatura by number86/goodfon.com

Lukke til med resten av eksamenssesongen, og tusen takk for eit fint semester!

Maks poeng: 20