

1(a)

```

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

```

Anta at kodesnutten over har blitt kjørt, og at en av setningene under er den neste setningen som utføres. Hva skrives ut? Hvis programmet krasjer, skriv kun Error.

(Husk at apostrofer og hermetegn som omgir strenger i kildekoden *ikke* er blir inkludert i utskriften.)

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)
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun 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)
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun 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)
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun Error)

Maks poeng: 3

- 1(e)** Skriv en funksjon **is_overweight** med to parametre **height** og **weight**. Du kan anta at «height» og «weight» viser til tallverdier for respektivt en persons høyde (i meter) og tyngde (i kilogram). Funksjonen skal returnere *True* hvis personen er overvektig, og *False* hvis ikke.

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

For å få uttelling på denne oppgaven må alle testene (inkludert hemmelige tester) passere.

Maks poeng: 4

- 1(f)** Skriv en funksjon **starts_with_h** med en parameter **animals**. Du kan anta at *animals* peker til en liste med strenger. Funksjonen skal returnere en ny liste som inneholder alle dyrene som begynner på bokstaven 'h'.

For å få uttelling på denne oppgaven må alle testene (inkludert hemmelige tester) 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 kjøres, krasjer det med feilmeldingen:

```
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 '='?

Deloppgave i:

- Forklar i korthet hva som er feil og hvordan programmet kan rettes.

Avhengig av hvilken tankefeil vi har gjort, er det ikke alltid forklaringsteksten i feilmeldingen og løsningsidéen den gir oss er spesielt hjelpsomme. Feilmeldingen sier likevel *noe* om hva som har gått galt.

Deloppgave ii:

- Forklar hvorfor det står «cannot assign to function call» i feilmeldingen.

Maksimalt 300 ord.

Skriv ditt svar her

Maks poeng: 10

2(b) Anta at du har en fil *foo.csv* med innholdet 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 kjører programmet som vises i vedlagte PDF. Hva blir innholdet i *bar.csv*?

Skriv ditt svar her

Maks poeng: 4

- 2(c)** Programmet i vedlagte PDF har dårlige variabelnavn, som gjør det vanskelig å lese og forstå. Forklar rollen til de ulike variablene i koden, og foreslå nye, selvbeskrivende navn for variabel- og funksjonsnavn.

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

Org. navn	Nytt navn	Forklaring av variabelens rolle
a		
b		
c		
d		
e		
f		
g		
h		

Skriv ditt svar her

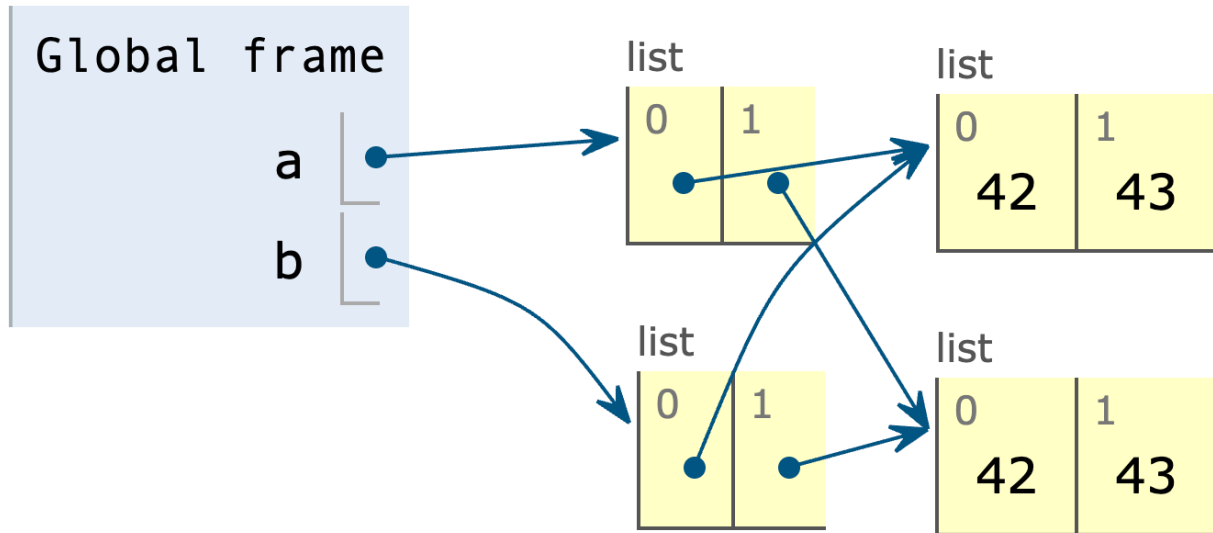
Maks poeng: 10

- 2(d)** Forklar hensikten med linje 17 i vedlagte PDF, og kom med et forslag til en bedre streng du kan bruke. «I love crashing!» er tross alt ikke særlig nyttig informasjon.

Skriv ditt svar her

Maks poeng: 4

3(a)



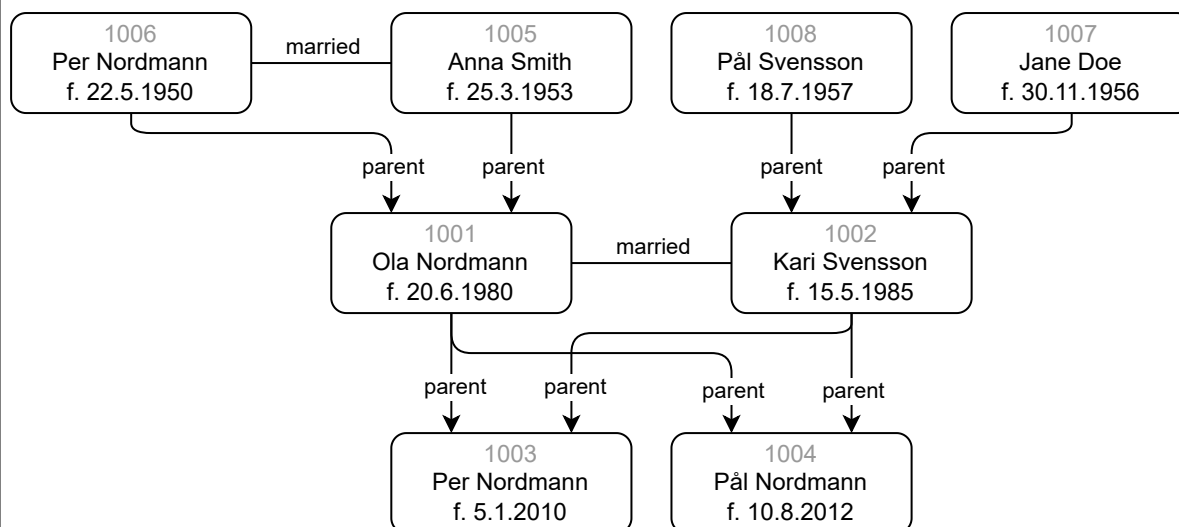
Skriv en kodesnutt slik at minnets tilstand blir som vist over. Du vil ikke få trekk i poeng dersom du definerer andre variabler i tillegg.

Skriv ditt svar her

Maks poeng: 3

3(b)

Ola og Kari har to sønner, Per og Pål, som er oppkalt etter besteforeldrene sine. Et lite utsnitt av familietreet deres ser du her:



I filen *family.json* vedlagt ser du en tekstuell representasjon av den samme informasjonen som på bildet. Hver person har en unik id. Id-nummeret har imidlertid ingen faktisk betydning utover å bli brukt for intern organisasjon av filen. For enkelhets skyld kan du anta at alle foreldreskap beskrevet i filen er biologiske; altså at adoptivforeldre og lignende relasjoner ikke er beskrevet.

Oppgavetekst

Ola har en tvilling, Nora Nordmann. Hvordan er det naturlig å utvide json-filen slik at Nora også blir med i familietreet?

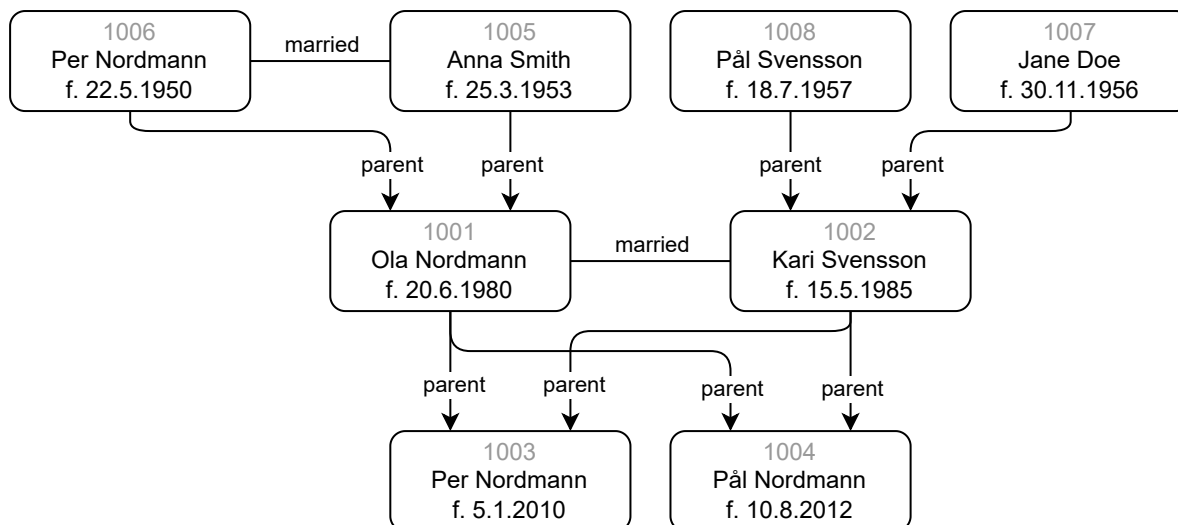
Du skal svare med å vise hvordan json-filen ser ut etter endringene dine. Du trenger ikke å kopiere hele json-filen, bare du viser hvordan den nye informasjonen er skrevet og viser/forklarer hvor den befinner seg i forhold til resten av innholdet i json-filen.

Skriv ditt svar her

Maks poeng: 4

3(c)

Ola og Kari har to sønner, Per og Pål, som er oppkalt etter besteforeldrene sine. Et lite utsnitt av familietreet deres ser du her:



I filen *family.json* vedlagt ser du en tekstuell representasjon av den samme informasjonen som på bildet. Hver person har en unik id. Id-nummeret har imidlertid ingen faktisk betydning utover å bli brukt for intern organisasjon av filen. For enkelhets skyld kan du anta at alle foreldreskap beskrevet i filen er biologiske; altså at adoptivforeldre og lignende relasjoner ikke er beskrevet.

Ola og Kari har blitt helt revet med, og legger til flere og flere familiemedlemmer. Til slutt har *family.json* blitt så stor at den inneholder informasjon om tusenvis av personer, samt foreldreskap og giftermål mellom dem.

Underveis i arbeidet legger de merke til at flere personer har akkurat samme navn; for eksempel heter Per Nordmann (født 2010) nøyaktig det samme som bestefaren sin (født 1950).

Oppgavetekst

Skriv en funksjon *get_ids* med to parametre:

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

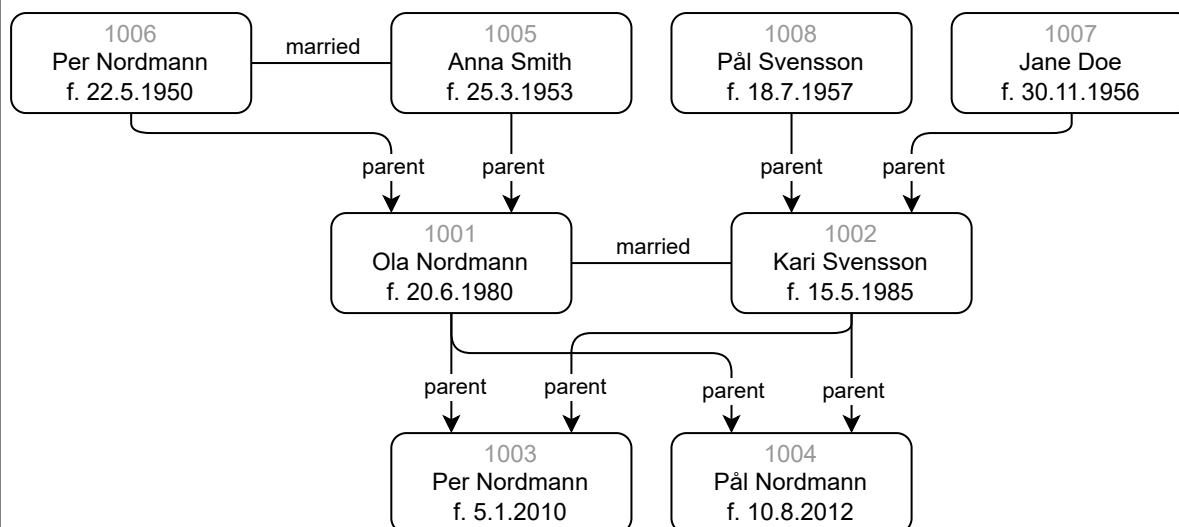
Funksjonen skal returnere en liste med id'er til personer med det gitte navnet. Eksempel: et kall til *get_ids(tree, 'Per Nordmann')* skal returnere en liste [1003, 1006].

Skriv ditt svar her

Maks poeng: 7

3(d)

Ola og Kari har to sønner, Per og Pål, som er oppkalt etter besteforeldrene sine. Et lite utsnitt av familietreet deres ser du her:



I filen *family.json* vedlagt ser du en tekstuell representasjon av den samme informasjonen som på bildet. Hver person har en unik id. Id-nummeret har imidlertid ingen faktisk betydning utover å bli brukt for intern organisasjon av filen. For enkelhets skyld kan du anta at alle foreldreskap beskrevet i filen er biologiske; altså at adoptivforeldre og lignende relasjoner ikke er beskrevet.

Ola og Kari har blitt helt revet med, og legger til flere og flere familiemedlemmer. Til slutt har *family.json* blitt så stor at den inneholder informasjon om tusenvis av personer, samt foreldreskap og giftermål mellom dem.

Oppgavetekst

Ola og Kari har en hypotese om at barn av ugifte foreldre (såkalte «kjærlighetsbarn») har større sannsynlighet for å bli født om våren (i mars, april eller mai). Skriv et program som finner

- hvor mange kjærlighetsbarn vi vet om totalt i slektstreet, og
- hvor mange av dem som ble født om våren.

Vi definerer et kjærlighetsbarn som en person hvor begge foreldre til personen er med i familietreet, men foreldrene ikke er markert som gift med hverandre.

Du kan anta at du har tilgang på en fil *family.json* på samme format som vist i eksempelet. I eksempelet finner vi ett kjærlighetsbarn (Kari), og hun ble født på våren (i mai). Koden du skriver må imidlertid virke også når *family.json* beskriver et slektstre med tusenvis av personer.

Du trenger ikke å ha optimal kjøretid for å få full uttelling på oppgaven. Det viktigste er at løsningen er korrekt og lett å forstå. Sensor vil legge vekt på at du klarer å dele opp oppgaven i egnede hjelpefunksjoner.

Skriv ditt svar her

Maks poeng: 14

- 4 I denne oppgaven blir det lagt inn poeng fra arbeidet ditt med laber du har gjort gjennom semesteret.



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

Lykke til med resten av eksamenssesongen, og tusen takk for et fint semester!

Maks poeng: 20