

1(a)

```

dukke = {
    1: 'bamse',
    'bamse': 'dukke',
    'dukke': 1,
    'smokk': 2,
}
smokk = 0

```

Anta at kodesnutten over har blitt kjørt. Hva skrives ut i disse setningene? Dersom programmet ville krasjet på linjen, skriv kun 'Error'.

| | |
|---|----------------------|
| <code>print(dukke[1][1])</code> | <input type="text"/> |
| <code>print(dukke['smokk'])</code> | <input type="text"/> |
| <code>print(1 < smokk)</code> | <input type="text"/> |
| <code>print(len(dukke))</code> | <input type="text"/> |
| <code>print(len('dukke'))</code> | <input type="text"/> |
| <code>print(dukke[dukke['dukke']])</code> | <input type="text"/> |
| <code>print(int("0") == 0)</code> | <input type="text"/> |
| <code>print(dukke[smokk])</code> | <input type="text"/> |

Maks poeng: 8
1(b)

```

a = 5
b = 3
a = a - b
b = a * b
a += 1
print(b - a)

```

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

Maks poeng: 5

1(c)

```
def f(y):  
    y -= 2  
    return y + 3  
def g(x):  
    x = f(x) + f(x)  
    return x + f(x)
```

Anta at funksjonene over er definert. Hva evaluerer disse uttrykkene til? (hvis uttrykket krasjer, skriv kun 'Error')

| | |
|---------|----------------------|
| f(2) | <input type="text"/> |
| g(2) | <input type="text"/> |
| f(g(1)) | <input type="text"/> |
| g(f(0)) | <input type="text"/> |

Maks poeng: 8

1(d)

```

def f(a, x):
    t = [a[0]]
    for r in a[1:]:
        if x in r[0]:
            t.append(r)
    return t
a = [
    ['name', 'age', 'section'],
    ['Adam', 930, 'rez'],
    ['Set', 912, 'bar'],
    ['Enosj', 905, 'bar'],
    ['Kenan', 910, 'rez'],
    ['Noah', 950, 'foo'],
]
b = f(a, 'e')

```

Anta at vi kjører koden over. Hva skrives ut dersom vi deretter kjører en av linjene under? (hvis programmet krasjer, skriv kun 'Error')

| | |
|------------------------------|----------------------|
| <code>print(a[2][1])</code> | <input type="text"/> |
| <code>print(b[2][1])</code> | <input type="text"/> |
| <code>print(b[1][-3])</code> | <input type="text"/> |

Maks poeng: 9
1(e)

```

a = [1, 0, 4]
t = 0
for x in a:
    t += 1
    for y in a:
        t += x * y
print(t)

```

Anta at vi kjører koden over. Hva skrives ut? (hvis programmet krasjer, skriv kun 'Error')

Maks poeng: 5

- 2(a)** Alfred skal plote en linje, og har skrevet et program over to Python-filer for å få det til. Når Alfred kjører `my_script.py` krasjer imidlertid programmet hans og viser en feilmelding i terminalen.

For å lese kildekoden til Alfred og feilmeldingen i terminalen, samt en illustrasjon av hvordan ønsket oppførsel skal være, se vedlagte PDF.

Forklar:

- Hva betyr feilmeldingen? Forklar hvilken informasjonen feilmeldingen gir oss.
- Hvordan kan du rette programmet slik at det viser et plot som vist i illustrasjonen i stedet for å krasje? Forklar nøyaktig hvor og hvordan du planlegger å gjøre rettelsen.

Bruk vanlig språk; du skal *ikke* skrive store kodesnutter. Samtidig må du være detaljert og presis nok til å overbevise sensor om at du forstår hva du snakker om.

Skriv ditt svar her

Maks poeng: 14

2(b)

```
def g(b, r):
    s = 0
    for e in b:
        if e == r:
            s += 1
    return s
def f(a):
    x = None
    for y in a:
        if x is None:
            x = y
        elif g(a, y) > g(a, x):
            x = y
    return x
```

Hva slags funksjoner er dette?

- Forklar hva funksjonene gjør.
- Koden over har dårlige variabel- og funksjonsnavn, som gjør den vanskelig å lese og forstå. Forklar rollen til de ulike variablene i koden over, og foreslå nye, selvbeskrivende navn for variabel- og funksjonsnavnene. Dersom du mener det ikke er nødvendig å endre et variabelnavn, forklar hvorfor variabelnavnet er egnet slik det allerede er.

8 poeng: forklar den øverste funksjonen, og gi bedre variabel- og funksjonsnavn til:

- g
- b
- r
- s
- e

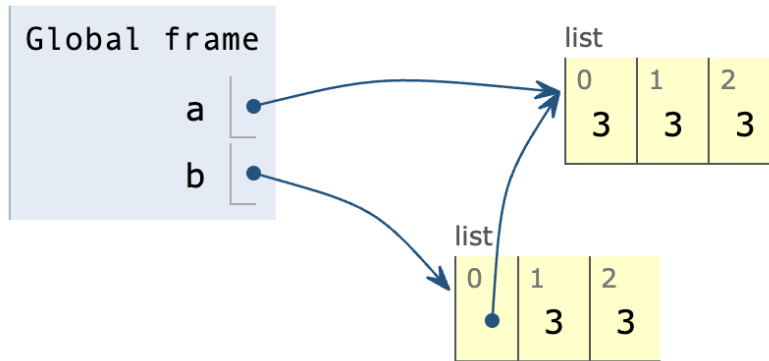
8 poeng: forklar den nederste funksjonen, og gi bedre variabel- og funksjonsnavn til:

- f
- a
- x
- y

Skriv ditt svar her

Maks poeng: 16

3(a)

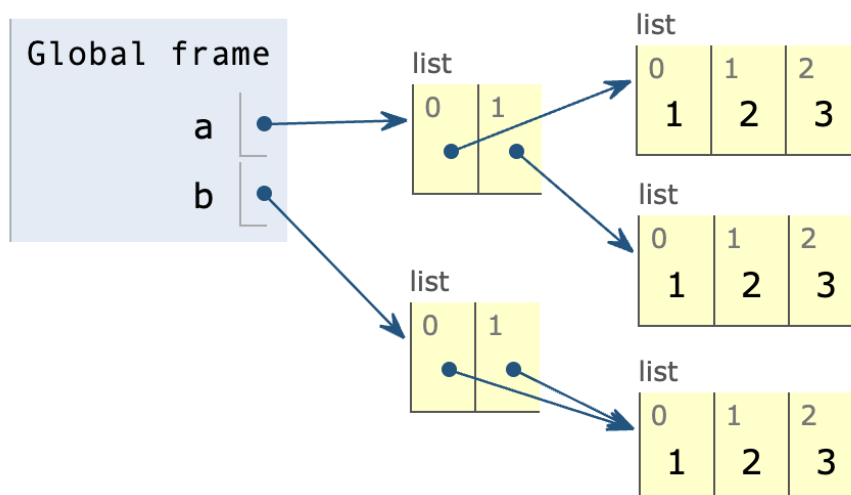


Skriv en kodesnutt slik at minnets tilstand blir som vist over. Du vil ikke få trekk i poeng dersom du definerer andre variabler i tillegg.

Skriv ditt svar her

Maks poeng: 5

3(b)



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

Skriv ditt svar her

Maks poeng: 5

3(c) Anta at du har en aktivitetslogg for et hotell. Aktivitetsloggen er en semikolon-separert csv-fil der hver rad representerer én hendelse. Kolonnene i csv-filen er

- tidspunkt (formattert som yyyy-MM-dd HH:mm)
- hendelsestype (enten «innsjekk» eller «utsjekk»)
- romnummer
- navn på gjesten

Eksempel på de første par linjene i filen:

```
tidspunkt;hendelsestype;romnummer;navn på gjest
2024-01-01 16:34;innsjekk;201;Donald
2024-01-01 17:15;innsjekk;302;Dolly
2024-01-02 09:14;utsjekk;201;Donald
2024-01-02 10:45;innsjekk;201;Magica fra Tryll
2024-01-02 10:59;utsjekk;302;Dolly
```

Anta at hendelsene allerede ligger i sortert rekkefølge basert på tidspunkt, og at ingen var sjekket inn på hotellet på tidspunktet loggen begynner. Du kan også anta at loggfilen er konsistent (utsjekk har alltid samme navn på personen som forrige gang det var innsjekk på samme romnummer; og det er alltid annenhver innsjekk og utsjekk om vi kun ser på ett rom av gangen).

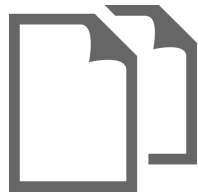
Det er tre deloppgaver:

- (5 poeng) Skriv en funksjon `load_logfile(path)` hvor `path` er en parameter for filnavnet til loggfilen. Funksjonen skal returnere innholdet i filen på et format egnet for videre prosessering i Python. Du kan velge selv om du vil returnere en 2D-liste eller en liste av oppslagsverk.
- (10 poeng) Skriv en funksjon `current_guest_count(table)` hvor `table` er en liste på det formatet du valgte i forrige deloppgave. Funksjonen skal returnere antallet gjester som er sjekket inn på hotellet når loggfilen tar slutt. *Eksempel: dersom filen kun bestod av linjene i eksempelet over, skal funksjonen returnere 1: fordi det er én gjest (Magica fra Tryll) som er sjekket inn når filen tar slutt.*
- (10 poeng) Skriv en funksjon `get_guests(table, time)` hvor `table` er en liste på det formatet du valgte i første deloppgave, og `time` er en streng som representerer et tidspunkt på samme format som brukes i loggfilen. Funksjonen skal returnere en liste over hvilke gjester som er sjekket inn og hvilke rom de befinner seg på. Mer konkret: funksjonen skal returnere en liste av tupler, der hver tuple består av (navn, romnummer) for en person som er sjekket inn på det gitte tidspunktet. *Eksempel: dersom filen kun bestod av linjene i eksempelet over, og det angitte tidspunktet er '2024-01-02 10:50' skal funksjonen returnere listen [('Magica fra Tryll', '201'), ('Dolly', '302')].*

Skriv ditt svar her

Maks poeng: 25

Question 6
Attached



my_lineplotter.py

```
1 | from matplotlib import pyplot as plt
2 |
3 | def plot_line(line):
4 |     xs = []
5 |     ys = []
6 |     for point in line:
7 |         x = point[0]
8 |         y = point[1]
9 |         xs.append(x)
10 |        ys.append(y)
11 |        plt.plot(xs, ys)
```

my_script.py

```
1 | from matplotlib import pyplot as plt
2 | from my_lineplotter import plot_line
3 |
4 | # Plot linjen (0, 0) -> (1, 2) -> (2, 1) -> (3, 3)
5 | line = [0, 0, 1, 2, 2, 1, 3, 3]
6 | plot_line(line)
7 | plt.show()
```

Terminal output

```
Traceback (most recent call last):
  File "/path/to/my_script.py", line 6, in <module>
    plot_line(line)
  File "/path/to/my_lineplotter.py", line 7, in plot_line
    x = point[0]
           ~~~~~^
TypeError: 'int' object is not subscriptable
```

Illustrasjon

