Informasjon

Eksamen består av tre deler, og gir tilsammen 100 poeng:

1. Automatisk rettede oppgaver (totalt 38 poeng)
2. Forklaringsoppgaver (totalt 20 poeng)
3. Kodeoppgaver (totalt 42 poeng)

Du bør ha begynt del 3 når du har 2 timer igjen. I del 3 vil du *ikke* ha anledning til å kjøre koden du skriver. Sensor er klar over dette, og vil ikke trekke poeng for småfeil som f. eks. skrivefeil i funksjonsnavn. Du må likevel skrive koden så tydelig og korrekt som mulig, slik at du demonstrerer at du forstår nyanser i koden du skriver.

Tillatte hjelpemidler på eksamen: alle skrivne og trykte hjelpemiddel.

Generelle råd

- Les spørsmålet før du svarer
- Jobb deg *raskt* gjennom alle spørsmålene i første runde, og kom heller tilbake til krevende oppgaver på nytt hvis du får tid på slutten.
- Hjelp sensor å hjelpe deg! Hvis du er usikker på tolkningen av en oppgave, gi en kort kommentar om hvordan du tolker usikkerheter i oppgaven. Hvis du ikke husker presist hvordan noe skal gjøres med kode, skriv en kommentar som forklarer hva du prøver på.

På denne siden finner du alle kursnotatene i emnet samlet i en PDF. De er tilgjengelig for deg dersom du ønsker. Det er helt frivillig å bruke dem. Vanlige regler for sitering gjelder: hvis du kopierer noe fra kursnotatene er det viktig at du oppgir dem som kilde i svaret ditt.

Du kan også finne kursnotatene nede i venstre hjørne gjennom hele eksamen. Da vil notatene åpnes i en ny fane.

1(a)

```
a = [-1, 3.14, 'foo']
b = 'bar'
c = 2
d = '[woz]'
e = {
  'foo': 2,
  0: [0],
  1: 'foo'
}
```

Anta at kodesnutten over har blitt kjørt. Hvilken datatype (klasse) får uttrykkene under hvis de evalueres?

2 + 2	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
a	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
d	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
a[c]	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
a[2][2]	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
'wax' in d	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
e[0]	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
e[-1]	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
len(e) / 2	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
c == (-2) * a[0]	

	Velg alternativ (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)

Maks poeng: 5

1(b)

```
a = [1, -1, 0, 2, 0, 4]
```

Gitt at koden over er kjørt.
Hva skrives ut i de følgende setningene? (hvis programmet krasjer, skriv kun 'Error')

print(a[1])	
print(a[2 + 2])	
print(a[3] + 1)	
print(a[a[-1]-1])	
print(a[a[a[0]]])	

Maks poeng: 5

1(c)

```
d = {
    'foo': 1,
    'bar': 2,
    0: 'foo',
    1: 0,
    2: 'woz',
    -1: -2
}
```

Gitt at koden over er kjørt.
Hva skrives ut i de følgende setningene? (hvis programmet krasjer, skriv kun 'Error')

print(d['foo'])	
print(d[2][0])	
print(d[d[0]])	
print('bar'[2])	

Maks poeng: 4

1(d)

```
x = 10
y = 5
x = x + y
y = x - y
x += 1
print(x + y)
```

Hva skriver dette programmet ut?

Maks poeng: 2

1(e)

```
a = [1, 3, 0, 4]
r = []
for i in range(len(a)):
    r.append(i)
    r.append(a[i])
print(f'{r[4]} {r[5]}')
```

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

Maks poeng: 2

1(f)

```
a = [6, 4, 3, -2, 4]
x = 10
for e in a:
    if e < x:
        x -= e
print(x)
```

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

Maks poeng: 2

1(g)

```
def bar():
    x = 20
    return x
x = 10
y = x
y = bar()
print(x + y)
```

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

Maks poeng: 2

1(h)

```
def baz(x):
    x += 2
    print(x, end='')
x = 5
baz(x)
print(baz(x))
```

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

Maks poeng: 2

1(i)

```
def quz(x, a):
    for i in a:
        x -= 1
    return x
p = 10
q = [10, 10, 10]
print(quz(p, q))
```

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

Maks poeng: 2

1(j)

```
def alpha(p, q):
    r = p + q
    s = beta(r + 1, q) + beta(p + 1, r)
    return s
def beta(t, u):
    v = t - u
    return v - 1
print(alpha(5, 2))
```

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

Maks poeng: 2

1(k)

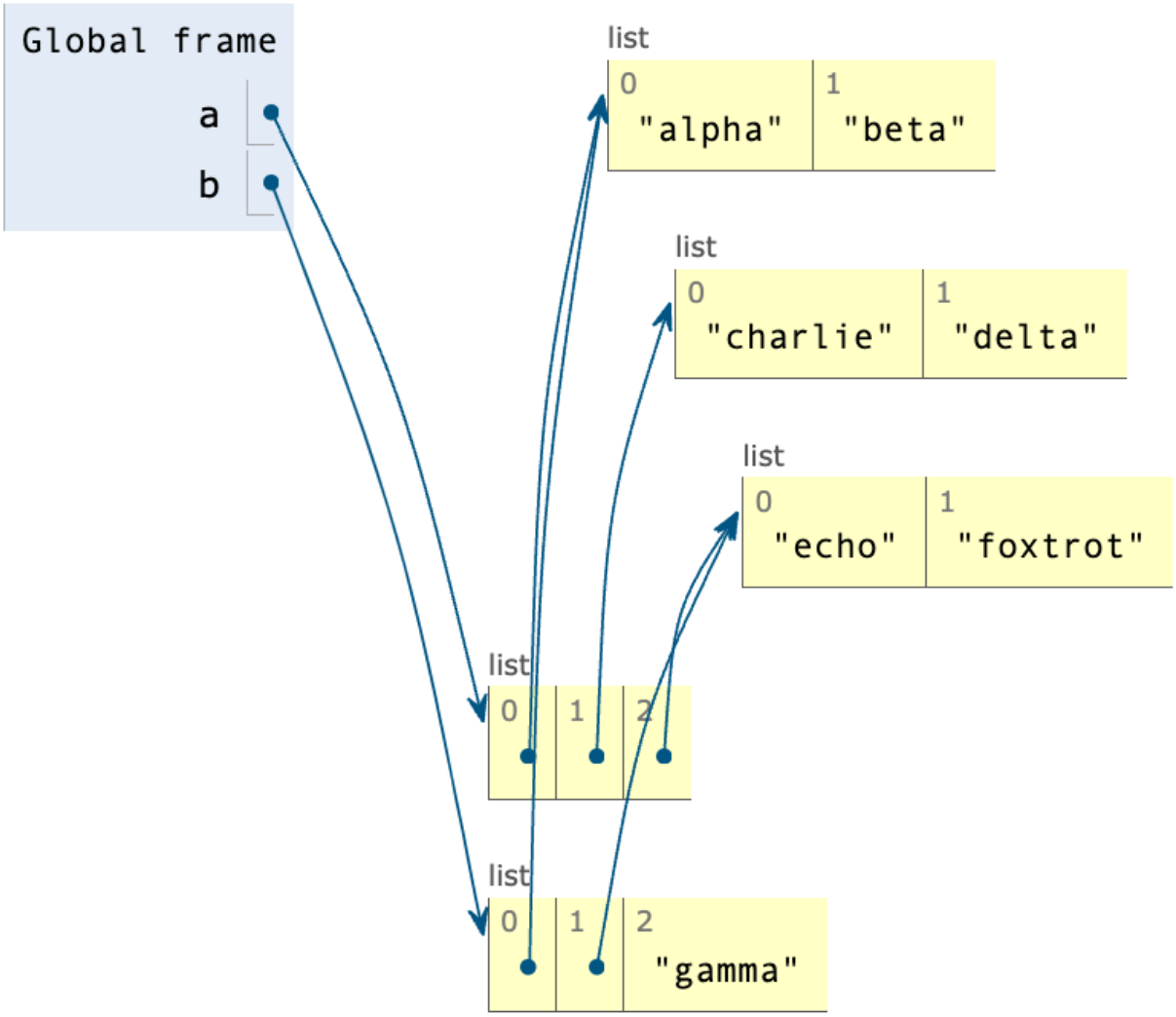
```
def echo(x):
    if x < 0:
        return 0
    if x >= 10:
        x = 10
    elif x % 2 == 0:
        x += 1
    if x < 5:
        x += 2
    else:
        x -= 1
    return x + 1
```

Gitt at funksjonen over er definert.
Hva skrives ut i de følgende setningene? (hvis programmet krasjer, skriv kun 'Error')

print(echo(0))	
print(echo(5))	
print(echo(10))	
print(echo(echo(2)))	

Maks poeng: 4

1(l)



Gitt at minnet har tilstanden vist over, hva blir skrevet ut etter setningen **print(a[1][1] + b[1][1])**? (hvis programmet krasjer, skriv kun 'Error')

Maks poeng: 2

1(m)

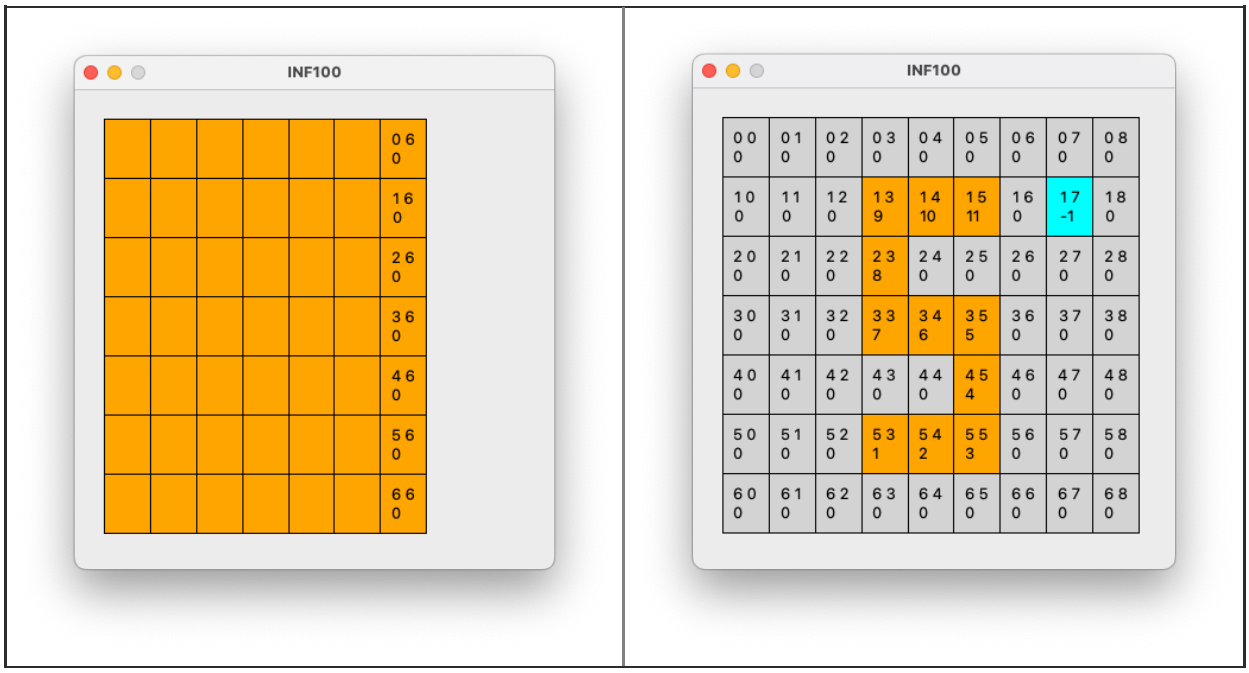
```
x = 999_8_22222_1_333_000_7777_66_44444444
b = 10
a = [0] * b
for i in range(b):
    t = x
    while t > 0:
        r = t % b
        if i == r:
            a[i] += 1
        t //= b
print(a[6])
```

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

PS: Programmet vil ikke krasje på første linje (det er lov å inkludere understreker «_» når man oppgir tall i Python for å gjøre det lettere å lese tallet).

Maks poeng: 4

2(a) Tidemann holder på med lab6 (Snake), men har gjort noe feil i steget der han skal tegne et rutenett. Når test-programmet hans *view_test.py* kjører, vises programmet til venstre, selv om han egentlig skulle ønske at det så ut som programmet til høyre. Det kommer ingen feilmeldinger i terminalen.



Hva har Tidemann gjort feil? Les koden hans til venstre, og forklar:

- hva han har gjort feil,
- hvorfor feilen fører til oppførselen som vises, og
- hva han kan gjøre for å rette feilene.

Skriv ditt svar her

Maks poeng: 10

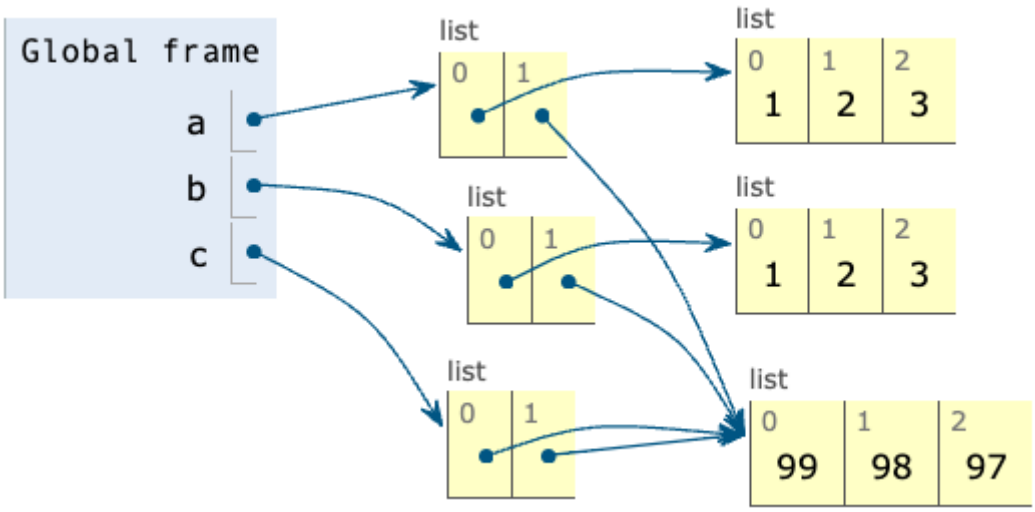
2(b) Othilie har lest seg opp om mengder og oppslagsverk, og har lagt merke til at de er *muterbare*. Men hva innbærer egentlig det? Hvilke hensyn må Othilie ta med muterbare verdier, som kanskje ikke ville vært nødvendig ellers?

Gi en forklaring til Othilie med illustrerende eksempler. Vi forventer ca 3-4 avsnitt, ikke mer enn 600 ord.

Skriv ditt svar her

Maks poeng: 10

3(a)



Skriv en kodesnutt slik at variabler og minnets tilstand for variablene a, b og c blir som vist over.

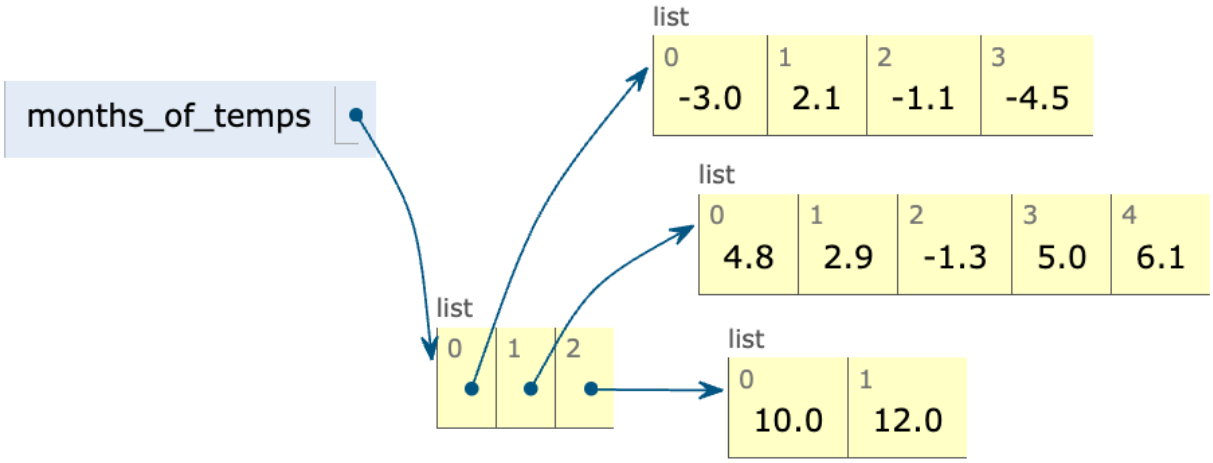
Skriv ditt svar her

Maks poeng: 4

3(b) Anta at **months_of_temps** er en variabel som peker på en to-dimensjonal liste av flyttall. De indre listene inneholder flyttall som representerer gjennomsnittstemperaturen for hver dag i en måned; mens den ytterste listen inneholder flere ulike måneder. Forskjellige måneder kan ha ulikt antall dager.

Skriv en funksjon **average_temp** med en parameter **months_of_temps** som beskrevet over. La funksjonen returnere gjennomsnittstemperaturen for alle dagene, uansett måned.

Eksempel på months_of_temps:



Hvis funksjonen du skriver kalles med eksempelet vist over som argument, skal returverdien bli 3.0: i eksempelet er det 11 ulike måleverdier (dager med temperaturmålinger), og summen av alle måleverdiene er 33.0.

Hint:

- tell hvor mange måleverdier som finnes,
- regn ut den totale summen av alle måleverdiene, og
- returner den totale summen delt på antall måleverdier.

Sensor vil vektlegge:

- Forståelse av løkker. Ikke bland indekser med verdier.
- Forståelse av funksjoner.
- Helhetlig algoritmisk forståelse.

Skriv ditt svar her

Maks poeng: 8

3(c) Oppgavetekst i PDF til venstre.
Skriv ditt svar her

Maks poeng: 20

3(d) Anta at den store Kvidoria har skrevet en ny episk roman i tekstfilen *story.txt*. Du jobber for forlaget, som er opptatt av et levende og variert språk. Redaktøren ber deg derfor om å lage et program som finner de 100 mest vanlige ordene i filen, samt hvor mange forekomster av disse ordene filen inneholder.

Hvilke hensyn må du ta for at løsningen skal fungere godt for redaktøren i praksis? *(Du skal ikke besvare dette som et forklaringsspørsmål, men la i stedet ditt svar på dette spørsmålet avgjøre hvilken funksjonalitet du inkluderer i programmet ditt.)*

Alle meningsfylte løsninger gir uttelling, men sensor vil vektlegge:

- anvendelighet for redaktøren,
- fornuftig valg av datastrukturerer, og
- hvor oversiktelig/lesbar koden er.

Skriv ditt svar her

Maks poeng: 10

Question 14
Attached

snake_view.py

```python
def draw_board(canvas, x1, y1, x2, y2, board, debug_mode):
    rows = len(board)
    cols = len(board[0])

    total_width = x2 - x1
    cell_width = total_width / cols
    total_height = y2 - y1
    cell_height = total_height / rows

    for row in range(rows):
        for col in range(rows):
            value = board[row][col]
            color = get_color(value)

            x_left = x1 + col * cell_width
            x_right = x_left + cell_width
            y_top = y1 + row * cell_height
            y_bottom = y_top + cell_height

            canvas.create_rectangle(x_left, y_top, x_right, y_bottom, fill=color)

        if debug_mode:
            xc = (x_left + x_right) / 2
            yc = (y_top + y_bottom) / 2
            text = f'{row} {col}\n{value}'
            canvas.create_text(xc, yc, text=text)

def get_color(value):
    color = None
    if value == 0:
        color = 'lightgray'
    if value >= 0:
        color = 'orange'
    if value == -1:
        color = 'cyan'
    return color
```

view_test.py

```python
from uib_inf100_graphics.simple import canvas, display
from snake_view import draw_board

test_board = [
    [0, 0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 9,10,11, 0,-1, 0],
    [0, 0, 0, 8, 0, 0, 0, 0, 0],
    [0, 0, 0, 7, 6, 5, 0, 0, 0],
    [0, 0, 0, 0, 0, 4, 0, 0, 0],
    [0, 0, 0, 1, 2, 3, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0, 0],
]

draw_board(canvas, 25, 25, 375, 375, test_board, True)
display(canvas)
```

Question 18
Attached

Geometrisk liste

En liste med tall kalles *geometrisk* hvis det finnes en konstant `k`, kalt *kvotienten*, slik at hvert tall, bortsett fra det første, er lik det foregående tallet multiplisert med `k`. For eksempel er `[3, 6, 12]` en geometrisk liste med kvotient `2`.

Forenklinger og antakelser for denne oppgaven:

- Geometriske lister med kvotient 0 er gørrkjedelige. En slik geometrisk liste består av kun 0'er, bortsett fra (kanskje) det første tallet. For enkelhets skyld definerer vi enhver liste som inneholder tallet 0 til å *ikke* være geometrisk.
- For enkelhets skyld sier vi også at en liste med færre enn to elementer er geometrisk med kvotient 1 (såfremt den ikke inneholder 0).

> Denne oppgaven består av fire deloppgaver (i), (ii), (iii), og (iv). Når du løser en gitt deloppgave kan du anta at alle funksjonen du skulle skrevet i de tidligere deloppgavene er korrekt implementert -- du står f. eks. fritt til å bruke funksjonen i oppgave (ii) som en hjelpefunksjon i oppgave (iii) selv om du ikke faktisk løste oppgave (ii).

Oppgave (i)

Skriv en funksjon `get_geometric_quotient(a)` som avgjør om en liste `a` med tall er geometrisk eller ikke. Hvis listen er geometrisk, returneres kvotienten. Hvis listen ikke er geometrisk, returneres None.

```
# Test 1
actual = get_geometric_quotient([3, 6, 12])
expected = 2
assert expected == actual

# Test 2
actual = get_geometric_quotient([4, 8, 8, 16, 3, 9, 27, 32])
expected = None
assert expected is actual

# Test 3
actual = get_geometric_quotient([42])
expected = 1
assert expected == actual
```

Oppgave (ii)

En **subliste** er en liste som består av en sammenhengende del av en annen liste. For eksempel:

- `[8, 8, 16]` er en subliste av `[4, 8, 8, 16, 4]`.
- `[4, 8, 16]` er *ikke* en subliste av `[4, 8, 8, 16, 4]`, fordi den ikke er sammenhengende/er ikke i samme rekkefølge.

Skriv en funksjon `len_geometric_sublist_starting_at(a, start_i)` som finner lengden til den lengste sublisten av `a` som starter på indeks `start_i` og som også er geometrisk.

```
a = [4, 8, 8, 16, 3, 9, 27, 81, 32]

# Test 1: fra i=2, er lengste subliste 8, 16 med k = 2, har lengden 2
actual = len_geometric_sublist_starting_at(a, 2)
expected = 2
assert expected == actual

# Test 2: fra i=3, er lengste subliste 16, 3 med k = 0.1875, har lengden 2
actual = len_geometric_sublist_starting_at(a, 3)
expected = 2
assert expected == actual

# Test 3: fra i=4, er lengste subliste 3, 9, 27, 81 med k = 3, har lengden 4
actual = len_geometric_sublist_starting_at(a, 4)
expected = 4
assert expected == actual
```

Oppgave (iii)

Skriv en funksjon `longest_geometric_sublist(a)` som returnerer den lengste sublisten av `a` som også er geometrisk. Dersom det er flere enn én delliste med denne lengden, returneres den tidligste av dem.

```
# Test
a = [4, 8, 8, 16, 3, 9, 27, 32]
actual = longest_geometric_sublist(a)
expected = [3, 9, 27]
assert expected == actual
```

Oppgave (iv)

Denne deloppgaven er en utfordring for dem som sikter på en A i emnet. Du bør spare denne til slutt, og løse de andre oppgavene på eksamen først.

> **I denne deloppgaven kan du *ikke* bruke modulen *itertools* om du vil ha full score**.
> Om du likevel benytter den, kan du maksimalt få 40% av poengene.

En **subsekvens** er en liste som består av en del av elementene i en annen liste, der elementene i subsekvensen opptrer i samme rekkefølge som i den opprinnelige listen. For eksempel:

- `[4, 16, 64]` er en subsekvens av `[4, 8, 8, 16, 64]`.
- `[4, 64, 16]` er *ikke* en subsekvens av `[4, 8, 8, 16, 64]`, fordi elementene ikke kommer i samme rekkfølge.

En subsekvens må ikke nødvendigvis være sammenhengende i den opprinnelige listen, men elementene må opptre i samme rekkefølge.

Skriv en funksjon `longest_geometric_subsequence(a)` som returnerer den lengste subsekvensn av `a` som også er geometrisk. Dersom det er flere enn én slik liste med denne lengden, skal du returnere den tidligste av dem.

```
# Test
a = [4, 8, 8, 16, 64, 3, 9, 27, 32]
actual = longest_geometric_subsequence(a)
expected = [4, 8, 16, 32]
assert expected == actual
```

Hint:

- Begynn med å gjette de første to elementene i subsekvensen.

Merk:

- Sensor retter først og fremst for korrekthet, ikke for effektivitet; men svært klumsete løsninger kan likevel bli trukket litt.