

i

# Egenerklæring

Jeg erklærer herved at besvarelsen som jeg leverer er mitt eget arbeid.

Jeg har ikke:

- samarbeidet med andre studenter
- brukt andres arbeid uten at dette er oppgitt
- brukt eget tidligere arbeid (innleveringer/ eksamenssvar) uten at dette er oppgitt

Om jeg har benyttet litteratur, vil en litteraturliste inneholde alle kilder jeg har brukt i besvarelsen og referanser vil vise til denne listen.

**Jeg er kjent med at brudd på disse bestemmelsene er å betrakte som fusk og kan føre til annullert eksamen og/eller utestengelse.**

Dersom du er usikker på om du kan stille deg bak erklæringen, se retningslinjer for bruk av kilder i skriftlige arbeider ved Universitetet i Bergen , og eventuelt ta kontakt med studieveileder/emneansvarlig.

Alle eksamensbesvarelser ved UiB blir sendt til manuell og elektronisk plagiatkontroll.

**Merk: Ved å fortsette bekrefter jeg at jeg har lest erklæringen og at besvarelsen jeg leverer under denne eksamenen er mitt eget arbeid (og bare mitt eget arbeid), i full overensstemmelse med ovennevnte erklæringen.**

i

# Informasjon

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

1. Automatisk rettede oppgaver (totalt 35 poeng)
2. Forklaringsoppgaver (totalt 30 poeng)
3. Kodeoppgaver (totalt 35 poeng)

I tredje del 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.

## 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.

1(a)

- a = "Hello, how are you!"
- b = { 3, 5, 1 }
- c = -1.4
- d = 64
- e = [ 42, "Bye", -3 ]

Velg datatype til uttrykket

	float	bool	int	(-error-)	list	str
c*d	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
len(b)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
f{b}'	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
[b]	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
b+d	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
a[e[2]]	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3 in e	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
a+a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Maks poeng: 8

1(b)

```

a = 2
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: 3

**1(c)**`a = [1, 0, -2, 2, 5, 3]`

Gitt at koden over er kjørt.

Hva skrives ut i de følgende setningene? (hvis programmet krasjer, skriv kun 'Error')

<code>print(a[4])</code>	<input type="text"/>
<code>print(a[3 - 1])</code>	<input type="text"/>
<code>print(a[3] - 1)</code>	<input type="text"/>
<code>print(a[a[0] - a[1]])</code>	<input type="text"/>
<code>print(a[a[a[0]]])</code>	<input type="text"/>

---

Maks poeng: 5

1(d) Velg slik at alle sammenligningene er True. Dict'et xs ser slik ut:

```
xs = {
  'a' : 5,
  '5' : 'hello',
  7 : 9.3781,
  5 : [7, 'Bergen', {3,5}],
}
```

Velg alternativ

'5' in

Velg alternativ

Velg alternativ

Maks poeng: 4

1(e) Velg verdien til dette boolske uttrykket:

a	b	c	(a and b) or c
True	False	False	<input type="text" value="Velg alternativ (True, False)"/>
True	True	False	<input type="text" value="Velg alternativ (False, True)"/>
True	False	True	<input type="text" value="Velg alternativ (True, False)"/>
False	False	True	<input type="text" value="Velg alternativ (False, True)"/>

Maks poeng: 4

- 1(f) Hint: bruk gjerne presedenstabellen i kursnotatene for å minne deg selv på hvilken operator som har presedens.

Hvordan plassere parenteser for å få et uttrykk *identisk* med

**12 // 2 \* 3**

**Velg ett alternativ**

- 12 // (2 \* 3)
- (12 // 2) \* 3

Hvordan plassere parenteser for å få et uttrykk *identisk* med

**x and y or z in a**

**Velg ett alternativ:**

- x and (y or (z in a))
- x and ((y or z) in a)
- (x and y) or (z in a)
- (x and (y or z)) in a
- ((x and y) or z) in a

1 poeng gis for hvert riktig svar, 0 poeng for feil svar eller ubesvart.

---

Maks poeng: 2

**1(g)**

```
def woz(s):  
    s += 'buzz'  
    print(s, end=' ')  
s = 'fizz'  
r = woz(s)  
print(f'{s} {r}')
```

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

---

Maks poeng: 2

**1(h)**

```
def qiz(x, a):  
    for e in a:  
        if x % 2 == 0:  
            x += e  
    return x  
q = [2, 4, 5, 6]  
print(qiz(0, q))
```

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

---

Maks poeng: 2

**1(i)**

```
def foxtrot(x):  
    if x >= 10:  
        return 10  
    else:  
        x += 10  
    if x > 10:  
        if x % 2 == 0:  
            x += 1  
        elif x >= 15:  
            x -= 1  
    else:  
        return 42  
  
    return x - 10
```

Gitt at funksjonen over er definert.

Hva skrives ut i de følgende setningene? (hvis programmet krasjer, skriv kun 'Error')

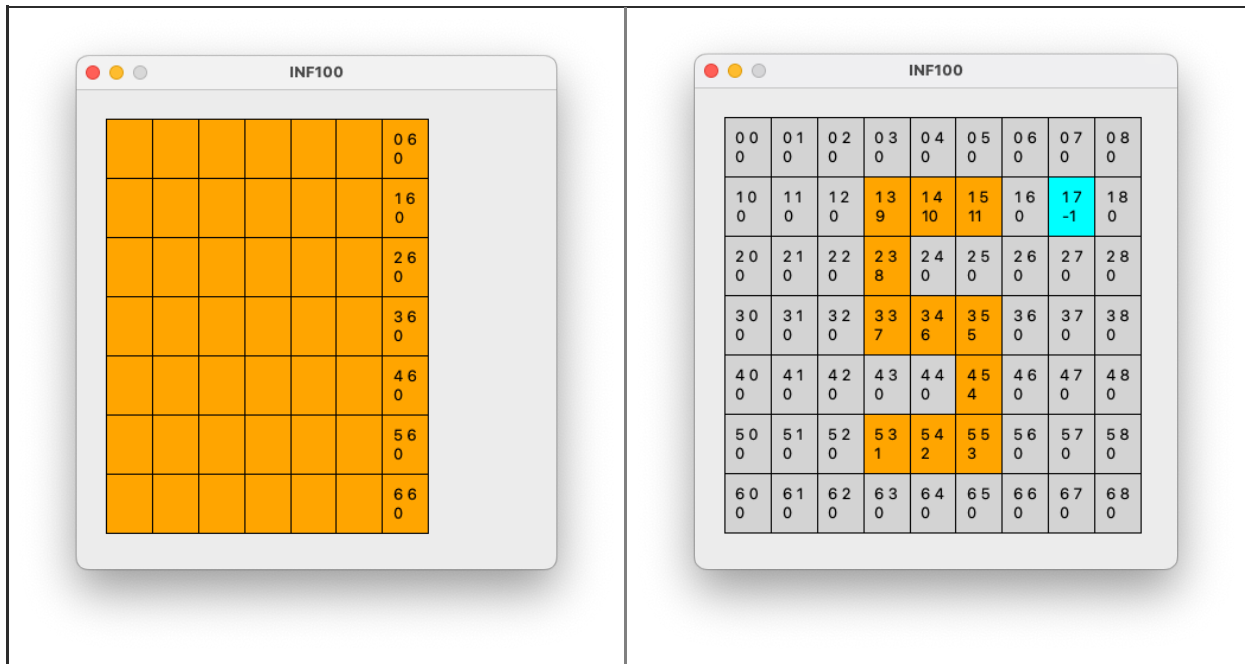
<code>print(foxtrot(0))</code>	<input type="text"/>
<code>print(foxtrot(2))</code>	<input type="text"/>
<code>print(foxtrot(3))</code>	<input type="text"/>
<code>print(foxtrot(5))</code>	<input type="text"/>
<code>print(foxtrot(foxtrot(2)))</code>	<input type="text"/>

---

Maks poeng: 5



- 2(a) Tidemann holder på med lab5 (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:

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

**Skriv ditt svar her**

Maks poeng: 10

2(b)

```
def count_a(s):
    count = {'a': 0}
    for c in range(len(s)):
        count[c] += 1
    return count['a']
```

Koden over skal telle hvor mange ganger tegnet "a" opptrer i en streng `s`, men det virker ikke. Finn alt som er feil, og beskriv hva man kan gjøre for å fikse funksjonen.

Ca to-tre avsnitt, helst ikke mer enn 200 ord.

**Skriv ditt svar her**

Maks poeng: 10

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

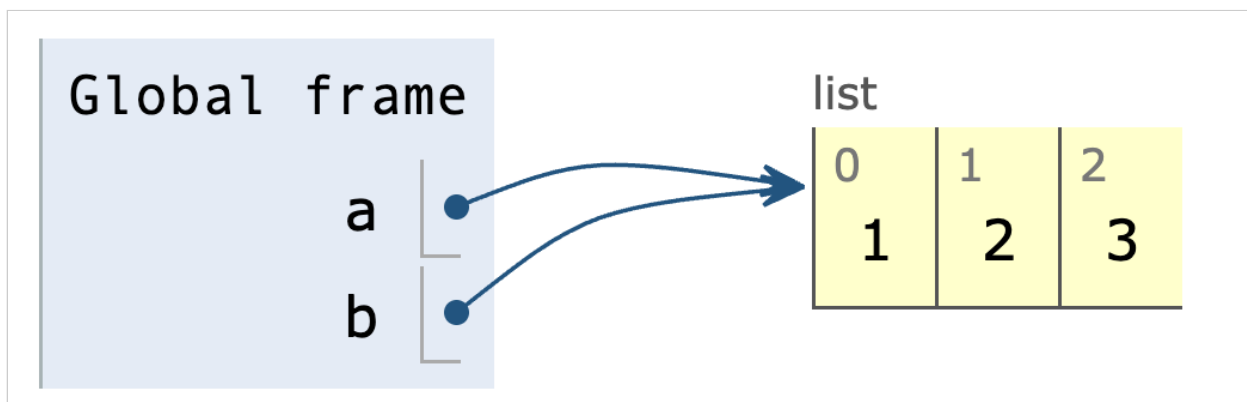
Gi en forklaring til Othilie med illustrerende eksempler, hvor eksemplene er basert på oppslagsverk. Vi forventer ca 3-4 avsnitt, ikke mer enn 600 ord.

**Skriv ditt svar her**

---

Maks poeng: 10

3(a)



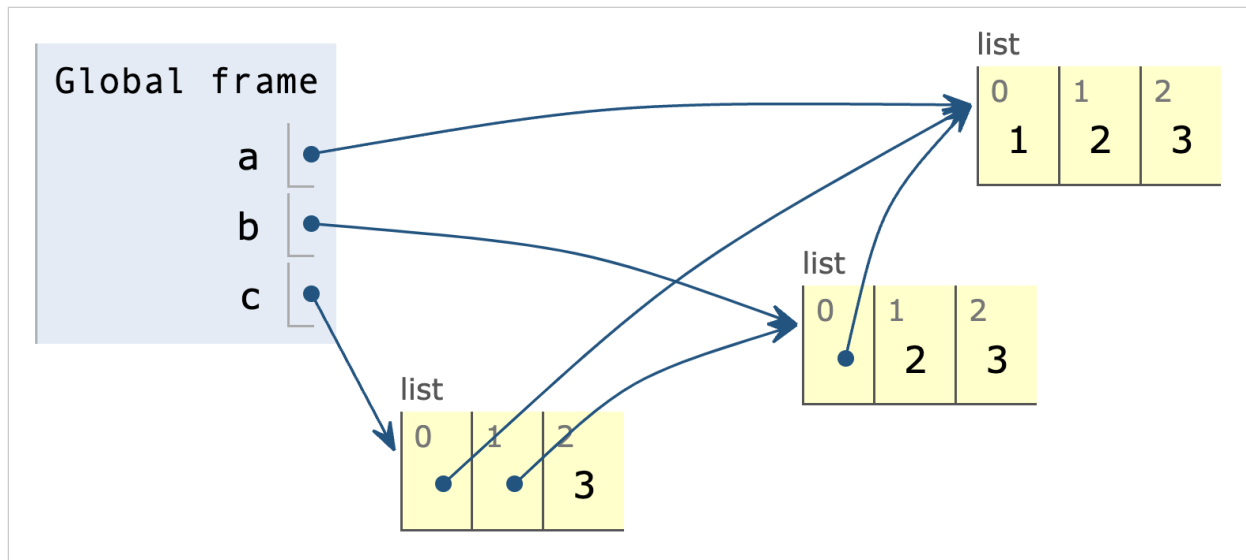
Opprett to variabler a, og b som refererer til den samme listen, slik at minnets tilstand blir som vist på bildet over.

**Skriv ditt svar her**

---

Maks poeng: 4

3(b)



Opprett tre variabler a, b og c, slik at minnets tilstand blir som vist på bildet over.

**Skriv ditt svar her**

---

Maks poeng: 6

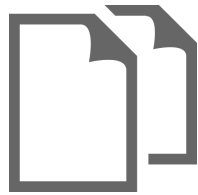
3(c)

**Skriv ditt svar her**

---

Maks poeng: 25

**Question 10**  
Attached



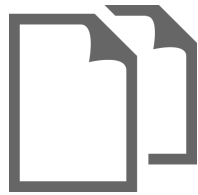
## snake\_view.py

```
1 def draw_board(canvas, x1, y1, x2, y2, board, debug_mode):
2     rows = len(board)
3     cols = len(board[0])
4
5     total_width = x2 - x1
6     cell_width = total_width / cols
7     total_height = y2 - y1
8     cell_height = total_height / rows
9
10    for row in range(rows):
11        for col in range(cols):
12            value = board[row][col]
13            color = get_color(value)
14
15            x_left = x1 + col * cell_width
16            x_right = x_left + cell_width
17            y_top = y1 + row * cell_height
18            y_bottom = y_top + cell_height
19
20            canvas.create_rectangle(x_left, y_top, x_right, y_bottom, fill=color)
21
22            if debug_mode:
23                xc = (x_left + x_right) / 2
24                yc = (y_top + y_bottom) / 2
25                text = f'{row} {col}\n{value}'
26                canvas.create_text(xc, yc, text=text)
27
28 def get_color(value):
29     color = None
30     if value == 0:
31         color = 'lightgray'
32     if value >= 0:
33         color = 'orange'
34     if value == -1:
35         color = 'cyan'
36     return color
```

## view\_test.py

```
1 from uib_inf100_graphics.simple import canvas, display
2 from snake_view import draw_board
3
4 test_board = [
5     [0, 0, 0, 0, 0, 0, 0, 0, 0],
6     [0, 0, 0, 9,10,11, 0,-1, 0],
7     [0, 0, 0, 8, 0, 0, 0, 0, 0],
8     [0, 0, 0, 7, 6, 5, 0, 0, 0],
9     [0, 0, 0, 0, 0, 4, 0, 0, 0],
10    [0, 0, 0, 1, 2, 3, 0, 0, 0],
11    [0, 0, 0, 0, 0, 0, 0, 0, 0],
12 ]
13
14 draw_board(canvas, 25, 25, 375, 375, test_board, True)
15 display(canvas)
```

**Question 15**  
Attached



# Nyttårsløpet 2017

I denne oppgaven skal du lage fem funksjoner som del av et rapporteringssystem fra et idrettsstevne «Nyttårsløpet 2017» som finner sted 31. desember 2017.

Funksjonene skal operere på en liste av deltakere. Hver deltaker i listen er igjen representert som en liste av 5 elementer: en streng som inneholder fornavn, en streng som inneholder etternavn, en bokstav som representerer kjønn ('K' for kvinne og 'M' for mann), et heltall som er fødselsåret, og en streng som representerer tiden vedkommende klarte å gjennomføre en halvmaraton på. ('1:59:20' betyr 1 time, 59 minutter og 20 sekunder.) Et eksempel på en slik liste av lister er som følger:

```
data = [
    ['Kari', 'Hansen', 'K', 1969, '1:59:20'],
    ['Eli', 'Nansen', 'K', 1975, '1:49:46'],
    ['Karl', 'Jansen', 'M', 1985, '1:35:40'],
    ['Erik', 'Karlsen', 'M', 1970, '1:40:48'],
    ['Anne', 'Jensen', 'K', 1964, '2:03:09'],
    ['Kurt', 'Johnsen', 'M', 1987, '1:32:43'],
    ['May', 'Berntsen', 'K', 1989, '1:36:54'],
    ['Jan', 'Thorsen', 'M', 1990, '1:45:24'],
    ['Hans', 'Monsen', 'M', 1998, '1:25:05'],
]
```

Som del av rapporteringssystemet skal du lage følgende fem Python-funksjoner.

- `group`
- `search`
- `winners`
- `time`
- `save_file`

Funksjonen `time` er en hjelpefunksjon til `winners`.

## group

Parametre:

- `data` en liste av lister som representerer resultatene fra et idrettsstevne som beskrevet over.
- `gender` en streng som enten er 'K' eller 'M'.
- `age_lower` en int som representerer nedre aldersgrense (inkludert)
- `age_upper` en int som representerer øvre aldersgrense (ekskludert)

Returverdi: *ingen*

Sideeffekter:

- Alle deltakere i `data` som har kjønn `gender` og som har alder mellom `age_lower` og `age_upper` på arrangementsdagen skal skrives ut på formatet `fornavn etternavn (alder) tid` på hver sin line.

Eksempelutskrift for et kall til `group(data, 'K', 25, 50)` gitt `data` som beskrevet over:

```
Kari Hansen (48) 1:59:20
Eli Nansen (42) 1:49:46
May Berntsen (28) 1:36:54
```

## search

Parametre:

- `data` en liste av lister som representerer resultatene fra et idrettsstevne som beskrevet over.
- `word` en streng som skal søkes etter.

Returverdi:

- en liste som representerer deltakerne fra `data` hvor `word` finnes i enten fornavnet eller etternavnet til deltakeren. Formatet på resultatlisten skal være akkurat den samme som for `data`, altså en liste av lister.

Sideeffekter: *ingen*

Eksempel på tester:

```
# Test 1
actual = search(data, 'Hans')
expected = [
    ['Kari', 'Hansen', 'K', 1969, '1:59:20'],
    ['Hans', 'Monsen', 'M', 1998, '1:25:05'],
]
assert expected == actual

# Test 2
actual = search(data, 'M')
expected = [
    ['May', 'Berntsen', 'K', 1989, '1:36:54'],
    ['Hans', 'Monsen', 'M', 1998, '1:25:05'],
]
assert expected == actual
```

## winners

Parametre:

- `data` en liste av lister som representerer resultatene fra et idrettsstevne som beskrevet over.

Returverdi: *ingen*

Sideeffekter:

- Den raskeste mannen og den raskeste kvinnen skal skrives ut på hver sin linje på formatet vist i eksempelet under.

Eksempelutskrift for et kall til `winners(data)` gitt `data` som i eksempelet over:

```
The fastest woman is May Berntsen with time 1:36:54
The fastest man is Hans Monsen with time 1:25:05
```

## time

Parametre:

- `s` en streng på formatet `H:MM:SS` som representerer en tid hvor `H` er antall timer, `MM` er antall minutter og `SS` er antall sekunder.

Returverdi:

- En int for antall sekunder som tilsvarer tiden `s`.

Sideeffekter: *ingen*

Dersom `s` ikke er på riktig format skal funksjonen krasje.

Eksempel på test:

```
actual = time("1:59:20")
expected = 7160
assert expected == actual
```

## save\_file

Parametre:

- `data` en liste av lister som representerer resultatene fra et idrettsstevne som beskrevet over.
- `filename` en streng som representerer filnavnet som resultatene skal lagres i.

Returverdi: *ingen*

Sideeffekter:

- Resultatene i `data` skal lagres i filen `filename` i et CSV-format hvor verdier separeres med semikolon. Første linje skal inneholde egnede overskrifter. De neste radene i filen skal inneholde fornavn, etternavn, kjønn, fødselsår og tid for hver av deltakerne.