

## Information

| Oppgave | Poeng | Oppgavetype                 |
|---------|-------|-----------------------------|
| i       |       | Informasjon eller ressurser |
| i       |       | Informasjon eller ressurser |
| i       |       | Informasjon eller ressurser |

## Autograded Tasks

| Oppgave | Poeng | Oppgavetype    |
|---------|-------|----------------|
| 1.1     | 3     | Fyll inn tekst |
| 1.2     | 5     | Paring         |
| 1.3     | 3     | Fyll inn tekst |
| 1.4     | 3     | Fyll inn tekst |
| 1.5     | 3     | Fyll inn tekst |
| 1.6     | 3     | Fyll inn tekst |
| 1.7     | 4     | Nedtrekk       |
| 1.8     | 3     | Flervalg       |
| 1.9     | 4     | Nedtrekk       |
| 1.10    | 4     | Fyll inn tekst |

## Explanation

| Oppgave | Poeng | Oppgavetype |
|---------|-------|-------------|
| 2.1     | 10    | Langsvar    |
| 2.2     | 10    | Langsvar    |
| 2.3     | 10    | Langsvar    |

## Code

| Oppgave | Poeng | Oppgavetype   |
|---------|-------|---------------|
| 3.1     | 4     | Programmering |
| 3.2     | 13    | Programmering |
| 3.3     | 18    | Programmering |

1.1

```
hammer = 0
tools = {
    0: 'drill',
    'box': {
        'handsaw': 1,
        'level': 4
    },
    (2, 4): 'box',
    'hammer': 2,
}
```

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

|   |                      |
|---|----------------------|
| <code>print("hammer")</code>              | <input type="text"/> |
| <code>print(tools[(2, 4)])</code>         | <input type="text"/> |
| <code>type(tools['box'])</code>           | <input type="text"/> |
| <code>print(len(tools))</code>            | <input type="text"/> |
| <code>print(tools[(2,4)] in tools)</code> | <input type="text"/> |
| <code>print(tools[hammer])</code>         | <input type="text"/> |

Maks poeng: 3

## 1.2

- a = 96
- b = "UiB"
- c = {3, 5, 2}
- d = -4.5
- e = [ 60, "Bergen", 5 ]

Velg datatype til uttrykket

|            | str                   | bool                  | float                 | int                   | list                  | -- Error --           |
|------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| a*d        | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| c[2]       | <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"/> |
| len(c)     | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| e[1][e[2]] | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| a*b        | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| 'UiB' 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"/> |
| [c]        | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| e+a        | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Maks poeng: 5

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

```
def foo(x):  
    x += 2  
    return x  
x = 10  
y = foo(x)  
print(x + y)
```

Svar:

---

Maks poeng: 3

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

```
def first_station_has_fuel(stations):  
    main_station = stations[0]  
    if main_station == 0:  
        return False  
    for station in stations[1:]:  
        if station == 1:  
            return False  
    return True  
print(not first_station_has_fuel([1, 0, 0, 1, 0]))
```

Svar:

---

Maks poeng: 3

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

```
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))
```

Svar:

---

Maks poeng: 3

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

```
def wut(x):  
    if x[0] == 'a':  
        if 'c' in x[2]:  
            if x[1] == 5:  
                return True  
            elif x[1] != 10:  
                y = str(x[1])  
                return 'y'  
        else:  
            if len(x) == 3:  
                y = x[1] + x[2]  
                return y  
            else:  
                return x[1]  
wut(['a', '123', 'bc'])
```

Svar:

---

Maks poeng: 3

1.7 Velg riktig svar fra en nedtrekksliste.

| a     | b     | c     | a or (b and c)                   | a or (not b)                     |
|-------|-------|-------|----------------------------------|----------------------------------|
| True  | False | False | Velg alternativ<br>(False, True) | Velg alternativ (True,<br>False) |
| True  | False | True  | Velg alternativ (True,<br>False) | Velg alternativ (True,<br>False) |
| False | True  | True  | Velg alternativ (True,<br>False) | Velg alternativ (True,<br>False) |
| False | False | True  | Velg alternativ<br>(False, True) | Velg alternativ<br>(False, True) |

Maks poeng: 4



1.8 Hvordan plassere parenteser for å få et uttrykk identisk med  $a \text{ or } b \text{ and } y < z \text{ or } c$

Velg ett alternativ:

- $((a \text{ or } b) \text{ and } (y < z)) \text{ or } c$
- $((((a \text{ or } b) \text{ and } y) < z) \text{ or } c$
- $((a) \text{ or } (b)) \text{ and } y < (z \text{ or } c$
- $a \text{ or } (b \text{ and } ((y < z) \text{ or } c))$
- $a \text{ or } (b \text{ and } (y < (z \text{ or } c)))$
- $(a \text{ or } (b \text{ and } (y < z))) \text{ or } c$

---

Maks poeng: 3

## 1.9 Velg et riktig svar.

|  |                                  |
|--|----------------------------------|
| <code>set('hello') ==<br/>{'e','l','o','o','h','h','h'}</code>           | Velg alternativ<br>(False, True) |
| <code>len([1, 2, 3] * 49) % 7 != 0</code>                                | Velg alternativ<br>(True, False) |
| <code>len([[[[['oops']]]]]) == 1</code>                                  | Velg alternativ<br>(True, False) |
| <code>True &lt; False</code>   | Velg alternativ<br>(True, False) |
| <code>[1].extend([2]) == [1].append(2)</code>                            | Velg alternativ<br>(True, False) |
| <code>len({'foo': 42}.values())<br/>&lt; len({'foo': 42}.items())</code> | Velg alternativ<br>(False, True) |
| <code>'pony' in 'toponymy'</code>  | Velg alternativ<br>(False, True) |
| <code>99 &lt; 1e2 &lt; 101</code>  | Velg alternativ<br>(False, True) |

Maks poeng: 4

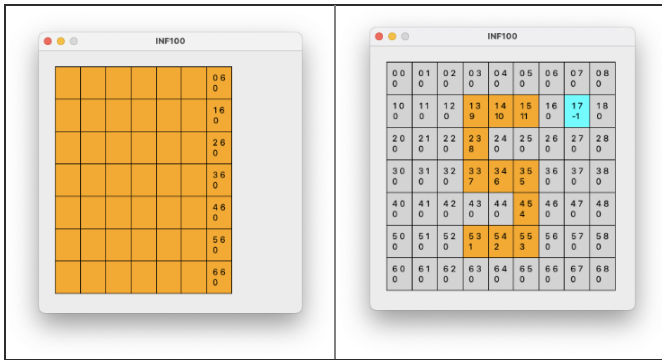
1.10 Fyll inn svarene dine i tabellen nedenfor:

|  |                      |
|--|----------------------|
| <pre>set([number + 1 for number in<br/>[0,1,1,2,3,5]])</pre>   | <input type="text"/> |
| <pre>''.join([let*2 for let in<br/>'hurra'])</pre>             | <input type="text"/> |
| <pre>[float(x) for x in '382475' if<br/>int(x) % 2 == 0]</pre> | <input type="text"/> |
| <pre>'lykke til'[::-1]</pre>                                   | <input type="text"/> |

```
set([number + 1 for number in [0,1,1,2,3,5]])
```

Maks poeng: 4

- 2.1 En student held på med lab5 (Snake), men har gjort noko feil i trinnet der et rutenett tegnes. Når testprogrammet `view_test.py` kjører, blir programmet til venstre vist, mens det skal se ut som programmet til høyre. Det kjem ingen feilmeldinger i terminalen.



Hva gikk galt? Les koden i pdf-filen og forklar:

- feilene i koden,
- hvorfor feilene fører til den oppførselen som vises, og
- hva som kan gjøres for å rette opp feilene.

Maks 400 ord.

**Skriv ditt svar her**

Maks poeng: 10

2.2

```
def count_s(string):  
    for c in range(len(string)):  
        count = 0  
        if c == "s":  
            count += 1  
    return count
```

Eliteserien er en norsk toppserie i fotball, og 17 forskjellige klubber har vunnet den minst én gang. Lucas skrev koden nedenfor for å telle hvor mange bokstaver 's' det er i strengen som inneholder Eliteserien-mestere. Av en eller annen grunn returnerer den imidlertid feil svar.

**Forklar:**

- Hvor gikk det galt i programutførelsen?
- Hvordan kan du korrigere programmet for å få det til å fungere? Forklar nøyaktig hvor og hvordan du har tenkt å gjøre korrigeringen.
- Hvilke spesialtilfeller av `str` input bør Lucas vurdere i det videre arbeidet med denne funksjonen?

Omtrent to til tre avsnitt, helst ikke mer enn 200 ord.

**Skriv ditt svar her**

---

Maks poeng: 10

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

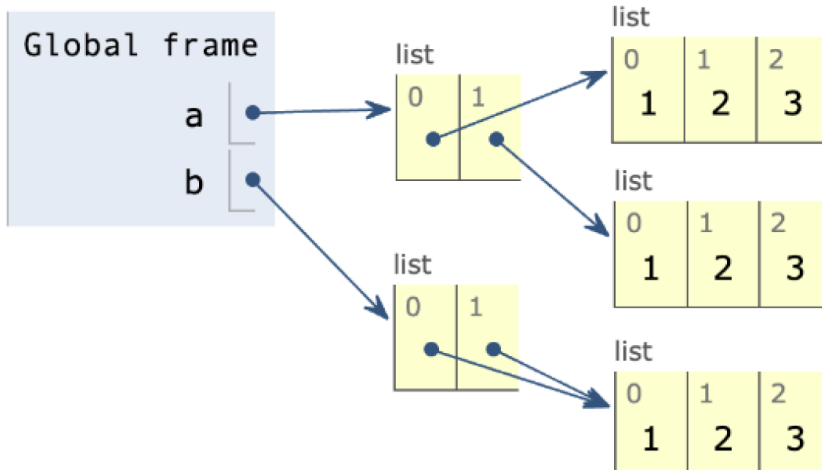
Gi en forklaring til Theo med illustrerende eksempler, hvor eksemplene er basert på oppslagsverk. Vi forventer ca 3-4 avsnitt, ikke mer enn 600 ord.

**Skriv ditt svar her**

---

Maks poeng: 10

3.1



Skriv en kodesnutt slik at variablene og minnets tilstanden blir som vist på bildet over. Du får ikke poengtrekk hvis du også definerer andre variabler.

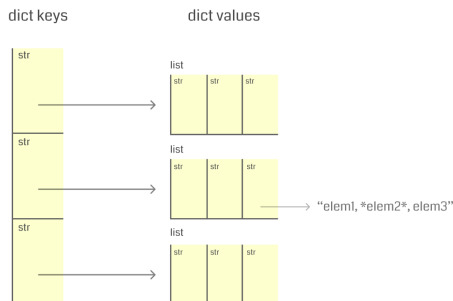
**Skriv ditt svar her**

Maks poeng: 4

## 3.2

```
mcu_films = {
  'Phase 1': ['*Iron Man*', 'The Incredible *Hulk*', '*Iron Man* 2', '*Thor*', '*Captain America*: The First Aver
  'Phase 2': ['*Iron Man* 3', '*Thor*: The Dark World', '*Captain America*: The Winter Soldier', '*Guardians of t
  'Phase 3': ['*Captain America*: Civil War', '*Doctor Strange*', '*Guardians of the Galaxy* Vol. 2', '*Spider-Me
}
```

Anta at `mcu_films` er en variabel som peker til en ordbok, der nøklene er fasene i Marvel Cinematic Universe' Infinity Saga-filmer (fra fase én til fase tre), og verdiene er lister over filmtitler som tilsvarer en bestemt fase. Filmens hovedkarakter er pakket inn i \*asteriskes\*. Eksempel på `mcu_films` er nedenfor; merk at antall filmer i hver fase er forskjellig.



**Oppgave:** Skriv en funksjon `count_protagonists` som tar data (dict beskrevet ovenfor) som en parameter. Funksjonen skal returnere ordboken over helter som opptrer som protagonister i MCUs Infinity Saga-filmer (på tvers av hele dict).

eksempel på utdata:

```
>>> count_protagonists(mcu_films)
{'Iron Man': 3, 'Hulk': 1, 'Thor': 3,..}
```

**Skriv ditt svar her**

---

Maks poeng: 13



## 3.3

```
1856.522; 1858.524; KIRK; How are the stone knives and bearskins?
1858.649; 1861.402; SPOCK; I may have found our focal point in time.
1861.527; 1862.319; KIRK; I think you may also find you have
1862.445; 1864.447; KIRK; A connect<<ERROR>>ion burning someplace.
1864.572; 1866.990; SPOCK; Yes. I'm overloading those lines.
1867.116; 1869.786; SPOCK; I believe we'll have our answer on this screen, Captain.
1869.910; 1870.911; KIRK; Good.
```

*Disse dataene kommer fra Star Trek: The Original Series, en banebrytende sci-fi-serie fra 1960-tallet som følger kaptein James T. Kirk og mannskapet hans om bord på USS Enterprise mens de utforsker verdensrommet og konfronterer fremmede sivilisasjoner. Selv om serien bare ble sendt i tre sesonger, ble Star Trek legendarisk og banet vei for fremtidig sci-fi og inspirerte til teknologi fra den virkelige verden. Kirks og Spocks eventyr kombinerte intens action med filosofiske temaer, noe som gjorde Star Trek til en hjørnestein i sci-fi-kulturen og en varig favoritt for fans som er interessert i verdensrommet, utforskning og samarbeid.*

Filen `startrek.csv` inneholder dialogdata hentet fra undertekstene til S01E28 «The City on the Edge of Forever». Hver rad i filen inneholder informasjon om når en replikk starter og slutter (i sekunder), hvilken karakter som snakker, og innholdet i dialogen. Et eksempel på en datastruktur finner du øverst.

#### Opgavene dine er

1. Skriv en funksjon `read_dialogue_data` som tar en parameterbane for filnavnet. Denne funksjonen skal lese innholdet i filen og returnere en 2D-liste, der hver subliste tilsvarer en rad i dataene. Hver rad skal deles med en spesifisert delimeter.
2. Skriv en funksjon `clean_dialogue_data` som søker etter og fjerner et spesifisert mønster `<<ERROR>>` i dialogteksten. Funksjonen skal (destruktivt) oppdatere dataene, fjerne alle forekomster av mønsteret og skrive ut indeksene til radene der mønsteret ble funnet.
3. Skriv en funksjon `calculate_average_duration` som beregner og returnerer ordboken for et gjennomsnittlig antall ord som snakkes per sekund for hvert tegn. Tips: Denne funksjonen bør bestemme:
  - Antall ord i hver linje,
  - Varigheten av hver linje (sluttidspunkt - startidspunkt), og
  - gjennomsnittlig antall ord per sekund for hvert tegn på tvers av alle linjene.

#### Skriv ditt svar her

---

Maks poeng: 18