

## Informasjon

Oppgåve	Poeng	Oppgåvetype
<b>i</b>		Informasjon eller ressurser
<b>i</b>		Informasjon eller ressurser
<b>i</b>		Informasjon eller ressurser

## Automatisk rettet

Oppgåve	Poeng	Oppgåvetype
1	4	Paring
2	4	Fyll inn tekst
3	6	Fyll inn tekst
4	3	Fyll inn tekst
5	6	Nedtrekk
6	4	Fyll inn tekst
7	3	Fleirval

## Forklaringsspørsmål

Oppgåve	Poeng	Oppgåvetype
8	5	Tekstfelt
9	5	Tekstfelt
10	5	Tekstfelt

## Kodingsspørsmål

---

<b>Oppg�ve</b>	<b>Poeng</b>	<b>Oppg�vetype</b>
11	5	Kompilering
12	5	Kompilering
13	10	Langsvar
14	15	Langsvar

**Poeng fra laber**

---

<b>Oppg�ve</b>	<b>Poeng</b>	<b>Oppg�vetype</b>
15	20	Munnleg

## **Egenerklæring**

Eg erklærer herved at bidraget eg sender inn, er mitt eige arbeid.

### **Eg har ikkje**

- samarbeidet med andre studentar
- brukt andre sitt arbeid utan å oppgi dette
- brukt eigne tidlegare arbeid (innleveringar/eksamenssvar) utan å oppgi dette

Om eg har brukt litteratur, skal ei litteraturliste innehalda alle kjeldene eg har brukt i oppgåva, og referansar skal visa til denne lista.

**Eg er klar over at brot på desse føresegnene blir rekna som fusk og kan føre til annullering av eksamen og/eller utestenging.**

Dersom du er usikker på om du kan stå inne for denne erklæringa, sjå retningslinjer for kjeldebruk i skriftlege arbeider ved Universitetet i Bergen, og ta kontakt med studierettleiaren/emnekoordinatoren din.

Alle eksamenssvar ved UiB blir sende til manuell og elektronisk plagiattkontroll.

**Merk: Ved å halde fram stadfestar eg at eg har lese erklæringa og at svaret eg leverer til denne eksamenen er mitt eige arbeid (og berre mitt eige arbeid), i fullt samsvar med erklæringa ovanfor.**

## Informasjon om eksamen

Eksamen består av tre delar, med til saman 80 poeng:

Type	Automatisk graderte spørsmål	Forklaringsspørsmål	Kodingsspørsmål
Poeng	30	15	35

- Dersom du har problem, kan du hoppe over eit spørsmål og gå tilbake til det seinare. For å unngå å gløyma spørsmål du har hoppa over, rår vi deg til å bruka flaggingsalternativet i øvre høgre hjørne.
- Du bør ha starta del 2 og 3 når du har 2,5 timar att.
- I del 3 vil du ha moglegheit til å køyre noko av koden du skriv. Men heile del 3 vil bli manuelt vurdert. Så du bør ikkje sitte fast i ei oppgåve når koden din ikkje består testtilfella. Eksaminator er klar over dette og vil ikkje trekkje poeng for mindre feil, for eksempel skrivefeil i funksjonsnamn frå standardbiblioteket. Du må likevel skrive koden så tydeleg og korrekt som mogleg for å vise at du forstår nyansane i koden du skriv.
- Tillatne hjelpemiddel til eksamen: alle skriftlege og trykte hjelpemiddel.

### Generelle råd

- Les spørsmålet før du svarar.
- Arbeid deg raskt gjennom alle spørsmåla i første runde, og kom tilbake til utfordrande spørsmål igjen dersom du har tid til slutt.
- Hjelp sensoren med å hjelpe deg! Dersom du er usikker på tolkinga av eit spørsmål, kan du skrive ein kort kommentar om korleis du tolkar usikkerheitsmomenta i spørsmålet. Dersom du ikkje hugsar nøyaktig korleis du skal gjere noko med kode, kan du skrive ein kommentar som forklarar kva du prøver å gjere.

**Materiell**

På denne sida finn du alle kursnotata i kurset samla i ei PDF-fil. Dei er tilgjengelege for deg om du ønskjer det. Det er heilt frivillig å bruke dei. Vanlege reglar for Sitering gjeld: Om du kopierer noko frå kursnotata, er det viktig at du oppgir dei som kjelde i svaret ditt.

Du kan òg finne kursnotata nedst i venstre hjørne under heile eksamen. Då opnar notata seg i ei ny fane.

PS: Har sidene plutseleg snudd seg sidelengs? Prøv å trykkje på «R» for å snu dei rett veg att.

1 `x = {'count':3}`  
`y = [8, x]`  
`w = ['strawberry']`

Vel riktig datatype for uttrykket

	dict	float	set	str	Error	int	bool	list
<code>len(w[0])</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>y[1]</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>f{123}</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>'brush' in 'toothbrush'</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>[42, True, x]</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>x['count']/2</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>{99, 40, 1}</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<code>x.pop(3)</code>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Maks poeng: 4

```
2 x = 10
  y = '10'
  u = [True, 2, 3]
  w = {
      'cba': 2,
      10: u,
  }
  s = 'cba'
```

Gå ut frå at kodesnutten over har vorte køyrt, og at ei av setningane under er den neste setninga som vert utført. Kva vert skrive ut? Dersom programmet krasjar, skriv berre Error.

(Hugs at apostrofar og hermeteikn som omgir strenger i kjeldekoden ikkje vert inkluderte i utskrifta.)

<code>print(x != y)</code>	<input type="text"/>
<code>print(not u[0])</code>	<input type="text"/>
<code>print(y + y)</code>	<input type="text"/>
<code>print(y[0])</code>	<input type="text"/>
<code>print(s[-1])</code>	<input type="text"/>
<code>print(u[1:])</code>	<input type="text"/>
<code>print(s*w[s])</code>	<input type="text"/>
<code>print(w[x][2])</code>	<input type="text"/>

Maks poeng: 4

3 def f(y):  
    y \*= 2  
    return y

def g(x):  
    x = f(x)  
    return f(x) + x

Gå ut frå at funksjonane over er definerte. Kva evaluerer desse uttrykka til? (dersom uttrykket krasjar, skriv berre 'Error')

f(3)	<input type="text"/>
g(4)	<input type="text"/>
f(g(1))	<input type="text"/>

Maks poeng: 6

```
4 def update(x, y):
    if x - y > 50:
        return 2*y
    elif x - y < 50:
        y += 50
    else:
        x -= 50
        print(y)
    return y // x
```

```
x = 100
y = x + 50
```

Gå ut frå at kodesnutten over har vorte køyrt, og at setninga under er den neste setninga som vert utført. Kva vert skrive ut? Dersom programmet krasjar, skriv berre Error.

```
print(update(x, y))
```

Maks poeng: 3

```
5 def f(x, y):
    result = []
    for num in x:
        if num % y == 0:
            result.append(num // y)
        else:
            result.append(num)
    return result
```

```
a = [8, 10, 36, 25]
```

```
b = f(a, 5)
```

Gå ut frå at kodesnutten over har vorte køyrt, og at ei av setningane under er den neste setninga som vert utført. Kva vert skrive ut? Dersom programmet krasjar, vel berre Error.

```
print(a)  ([8, 2, 36, 25], [8, 10, 36, 25], Error, [8, 10, 36, 5], [8, 2, 36, 5])
```

```
print(b)  ([8, 2, 36, 25], [8, 2, 36, 5], [8, 2.0, 36, 5.0], [8, 10, 36, 5], Error)
```

```
print(f(f(a, 5), 5))  ([8, 2, 36, 1], Error, [8, 1.0, 36, 2.0], [8, 1, 36, 2], [8, 2.0, 36, 1.0])
```

Maks poeng: 6

```
6 x = 20
  y = 1
  while x > 0:
    x -= 2
    if x % 4 == 0:
      continue
    if x < 10:
      break
    y = 1 - y
```

Gå ut frå at kodesnutten over har vorte køyrt, og at setninga under er den neste setninga som vert utført. Kva vert skrive ut? Dersom programmet krasjar, skriv berre Error.

```
print(x+y) 
```

Maks poeng: 4

- 7 Denne oppgåva handlar om presedens og assosiativitet.  
Korleis plassere parentesar for å få eit uttrykk identisk med  
**x or not y and z or w**

**Vel eitt alternativ**

- x or (((not y) and z) or w)
- (x or ((not y) and z)) or w
- x or ((not (y and z)) or w)
- (x or (not (y and z))) or w

Maks poeng: 3

- 8 Ei revolusjonær nemnd har teke makta. Dei gav i oppdrag å utvikle følgjande program for å administrere leiarskapen:

```
def foo(lst):
    last = lst[-1]
    for i in range(len(lst) - 1, 0, -1):
        lst[i] = lst[i - 1]
    lst[0] = last
```

```
names = ["Bashar", "Robert", "Fidel", "Vladimir", "Kim"]
foo(names)
print(names)
```

- Kva gjer funksjonen foo()? Kva skriv programmet ut?
- Gje funksjonen foo() eit betre namn, slik at folk enkelt kan forstå kva funksjonen gjer utan å vite korleis han er implementert.

**Skriv svaret ditt her**

Maks poeng: 5

- 9 Storleiken på eit embryo blir målt i dei første vekene av svangerskapet ved hjelp av CRL (crown-rump length) – avstanden frå toppen av hovudet til enden av ryggraden. Ein lege ønskjer å estimere CRL-verdiane for neste trimester basert på ein forventa vekstfaktor. Følgjande funksjon blir brukt til å rekne ut dette:

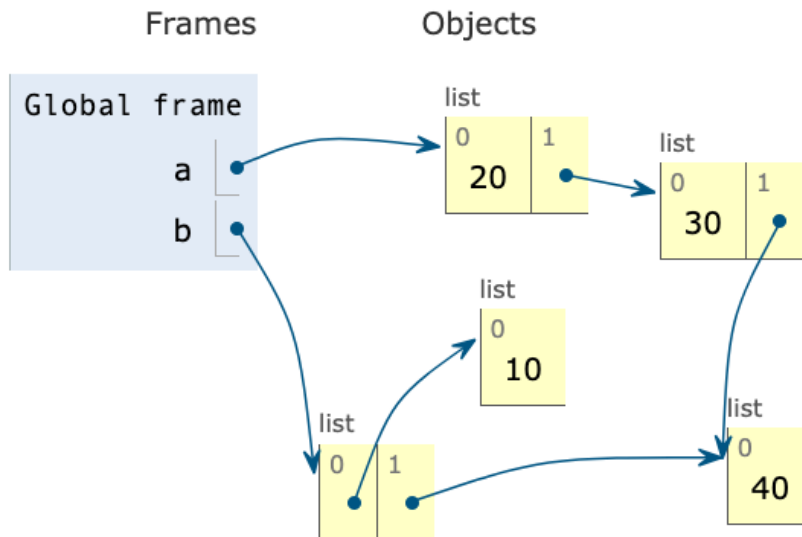
```
def times_factor(crl_measurements, factor):
    new_lst = []
    for sub_lst in crl_measurements:
        new_sub_lst = []
        for num in sub_lst:
            new_sub_lst.append(num*factor)
        new_lst.append(new_sub_lst)
    return new_lst
```

```
data = [[12, 16, 21], [38, 45, 52]]
times_factor(data, 10)
print(data)
```

- Køytinga gir outputen [[12, 16, 21], [38, 45, 52]], som ikkje var venta. Kva trur du den venta utskrifta burde vere?
- Kvifor klarer ikkje programmet å gjere dette?
- Korleis vil du fikse det?

**Skriv svaret ditt her**

Maks poeng: 5



Skriv ein kodesnutt slik at minnet får den tilstanden som er vist ovanfor. For full pott må du løyse oppgåva utan å bruke hjelpevariablar/andre variablar enn **a** og **b**. (bilete er henta frå Python Tutor).

**Skriv svaret ditt her**

Maks poeng: 5

- 11 Skriv ein funksjon `count_letter(letter, wordlist)` som tel kor mange gonger ein gitt bokstav førekjem i ei liste med ord. Til dømes førekjem bokstaven 'b' totalt 5 gonger i lista ['stubborn', 'house', 'cabbage', 'job', 'exam'].

Skriv svaret ditt her

Test case #	Input	Forventa output
1	<code>count_letter('b', ['stubborn', 'house', 'cabbage', 'job', 'exam'])</code>	5
2	<code>count_letter('k', ['bookkeeping', 'lake', 'kickoff', 'tree', 'rucksack'])</code>	7

```
def count_letter(letter, wordlist):  
    # Skriv koden her  
    # ...
```

```
# IKKE rediger under denne linjen!
```

```
print(eval(input()))
```

```
|
```

Test kode

Maks poeng: 5

12 Halt fram med den førre oppgåva (Oppgåve 11).

Skriv ein funksjon `count_letters(data)`.

- `data` er ei liste med tuplar på forma (letter, wordlist)

Funksjonen skal returnere ei liste med heiltal, der kvart tal viser kor mange gonger den tilhøyrande bokstaven førekjem i ordlista si.

**Skriv svaret ditt her. Endringer blir lagret automatisk.**

Test case #	Input	Forventa output
1	<code>count_letters([('b', ['stubborn', 'house', 'cabbage', 'job', 'exam']), ('k', ['bookkeeping', 'lake', 'kickoff', 'tree', 'rucksack'])])</code>	[5, 7]

```
def count_letters(data):  
    # Skriv koden her  
    # ...  
  
# IKKE rediger under denne linjen!  
print(eval(input()))
```

Test kode

Maks poeng: 5

13

Skriv eit Python-program som les ei CSV-fil med personopplysningar, og rekner ut det tilrådde daglege vassintaket for kvar enkelt person (i milliliter).

**Vektbasert (metrisk):** Multipliser vekta (kg) med 30 (ml/kg).

(Til dømes:  $70 \text{ kg} * 30 \text{ ml/kg} = 2100 \text{ ml per dag}$ )

Gå ut frå at du har ei fil *health.csv* med innhaldet nedanfor:

```
Name;Weight(kg)
Henrik;70
Ingrid;52
Mathias;85
Svanhild;60
```

Utskrifta frå programmet ditt bør vere ei liste med oppslagsverk:

```
[{"Name": "Henrik", "Weight(kg)": 70, "Water": 2100},
 {"Name": "Ingrid", "Weight(kg)": 52, "Water": 1560},
 {"Name": "Mathias", "Weight(kg)": 85, "Water": 2550},
 {"Name": "Svanhild", "Weight(kg)": 60, "Water": 1800}]
```

**Skriv svaret ditt her**

Format | ↺ | ↻ | ✎ | Σ | ✖

Words: 0

Maks poeng: 10



Words: 0

Maks poeng: 15

```
{
  "1" : { "planned_idx": [...], "actual_idx": [...], "distances": [...] },
  "2" : {
    "planned_idx": [
      0,
      1,
      2
    ],
    "actual_idx": [
      0,
      2,
      1
    ],
    "distances": [
      [
        0,
        30870,
        31775
      ],
      [
        31004,
        0,
        1499
      ],
      [
        32054,
        1625,
        0
      ]
    ]
  },
  ...,
  "100" : { "planned_idx": [...], "actual_idx": [...], "distances": [...] }
}
```

**15** Her blir poenga dine frå labbane lagde til.

Maks poeng: 20