

## Flere oppgaver

Her finner du flere relevante oppgaver – for alle de ulike temaene. Dersom du blir fort ferdig, ønsker å øve deg mer på programmering eller bare ønsker inspirasjon til oppgaver du selv kan gi til dine elever igjen kan du gjerne prøve deg på disse.

Oppgavene her er ganske forskjellige i omfang og vanskelighetsnivå. Noen er rettet mot grunnleggende programmering, som er gode hvis man vil repetere ulike konsepter. Andre er mye mer utfordrende, men vi håper at de også er mer interessante. Det kan lurt å være bevisst på hva man vil fokusere på – mengdetrening eller utfordringer. Plukk gjerne oppgaver som er relevant for de fagene du selv underviser i.

God koding!

# Innhold

<b>1</b>	<b>Variabler og regning</b>	<b>3</b>
1.1	Grunnleggende programmering . . . . .	3
1.2	Matematikk . . . . .	3
1.3	Kjemi . . . . .	9
1.4	Fysikk/biologi . . . . .	10
<b>2</b>	<b>Betingelser</b>	<b>12</b>
2.1	Grunnleggende programmering . . . . .	12
2.2	Matematikk . . . . .	13
2.3	Naturfag . . . . .	19
<b>3</b>	<b>Løkker</b>	<b>21</b>
3.1	Grunnleggende programmering . . . . .	21
3.2	Matematikk . . . . .	22
3.3	Naturfag . . . . .	25
<b>4</b>	<b>Funksjoner</b>	<b>27</b>
4.1	Grunnleggende programmering . . . . .	27
4.2	Matematikk . . . . .	27
4.3	Naturfag . . . . .	28
4.4	Fysikk . . . . .	29
<b>5</b>	<b>Plotting</b>	<b>32</b>
5.1	Matematikk . . . . .	32
5.2	Naturfag . . . . .	34

# 1 Variabler og regning

## 1.1 Grunnleggende programmering

### Oppgave 1 *Tolke feilmedlinger*

I denne oppgaven blir du presentert for noen av de vanligste feilmeldingene man kan få når man programmerer. Her trenger du ikke programmere noe, men les meldingen nøye, og beskriv hva du tror kan føre til en slik feilmelding. Finn også ut hvilken linje feilen er på.

a)

```
Traceback (most recent call last):  
  File "test.py", line 6, in <module>  
    print(f'Jeg er {alder} år gammel!')  
NameError: name 'alder' is not defined
```

b)

```
Traceback (most recent call last):  
  File "test.py", line 4, in <module>  
    sum = tall1 + tall2  
TypeError: unsupported operand type(s) for +: 'int'  
         ' and 'str'
```

c)

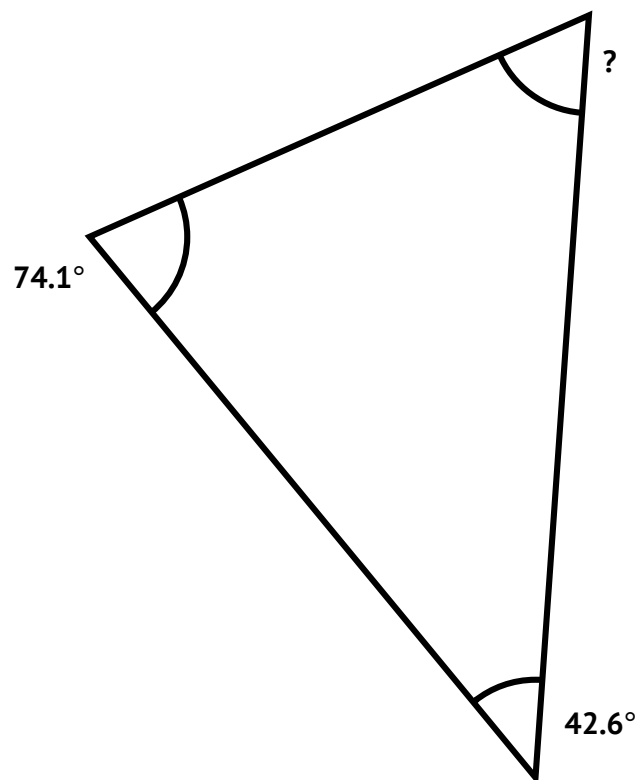
```
File "test.py", line 3  
    navn = 'Jonas  
           ^  
SyntaxError: EOL while scanning string literal
```

## 1.2 Matematikk

### Oppgave 2 *Vinkelsum*

Vinkelsummen til en trekant er alltid  $180^\circ$ . I denne oppgaven skal du lage et program som regner ut den siste vinkelen, gitt at du kjenner til den første vinkelen.

- a) Lag en variabel `første_vinkel` og sett den til å være  $90^\circ$
- b) Lag en variabel `andre_vinkel` og sett den til å være  $60^\circ$
- c) Regn ut hva den tredje vinkelen i trekanten må være og lagre svaret i en variabel `tredje_vinkel`
- d) Bruk programmet ditt til å finne størrelsen på den ukjente vinkelen i figuren under:



### Oppgave 3 *Radianer*

Får å regne om fra grader til vinkler kan vi bruke følgende formel:

$$\theta = \frac{\pi \phi}{180},$$

Hvor  $\theta$  er vinkelen i radianer og  $\phi$  er vinkelen i grader

- a) Lag en variabel `vinkel_grader` og gi den verdien 45.
- b) Regn ut vinkelen i radianer og lagre det i en variabel `vinkel_radianer`. Skriv ut hvor mange radianer som tilsvarer  $45^\circ$
- c) Modifiser programmet ditt så det spør brukeren om en vinkel i grader skriver ut hva det er i radianer.
- d) Finn et uttrykk for å gjøre om fra radianer til grader og lag et program som ber om en vinkel i radianer og skriver det ut i grader.

#### Oppgave 4 *Regne mellom SI-enheter*

En millimeter er 0.01 centimeter. En mikrometer er 0.001 millimeter. En centimeter er 10 000 mikrometer.

Lag en variabel med din høyde i cm, og lag en ny variabel som gjør denne høyden til mm. Lag enda en ny variabel som gjør høyden i mm til  $\mu\text{m}$ . Til slutt, lag en variabel som på ny definerer din høyde i cm, men regnet fra  $\mu\text{m}$ . Print denne siste variabelen, har du regnet rett og fått riktig høyde i cm?

#### Oppgave 5 *Konvertering av temperatur*

I Norge oppgir vi temperaturer i målestokken *Celsius*, men i USA bruker de ofte målestokken *Fahrenheit*. Hvis du finner en kakeoppskrift fra USA kan det for eksempel stå at du skal bake kaken ved 350 grader. Da mener de altså  $350^\circ\text{F}$ . Vi vil nå lage et verktøy som kan konvertere denne temperaturen for oss, sånn at vi vet hva vi skal bake kaken ved i Celsius..

For å regne over fra Fahrenheit til Celsius bruker vi formelen:

$$C = \frac{5}{9}(F - 32).$$

Der  $F$  er antall grader i Fahrenheit, og  $C$  blir antall grader i celsius.

- a) Lag et program som spør brukeren om en temperatur i antall grader Fahrenheit, og skriver ut den tilsvarende temperaturen i antall grader Celsius.

Husk å gjøre om svaret til et tall, med enten `int(input())` eller `float(input())`.

- b) Bruk programmet ditt til å finne ut hvor mange grader Celsius  $350^{\circ}\text{F}$  svarer til. Virker det rimelig å skulle bake en kake ved denne temperaturen?

Programmet du har lagd tar en temperatur i Fahrenheit, og gjør om til Celsius. Men hva om vi ønsker å gå motsatt vei? Om vi ønsker å lage et nytt program som gjør motsatt, så må vi først ha en formel for  $F$ .

- c) Klarer du å ta uttrykket

$$C = \frac{5}{9}(F - 32).$$

og løse for  $F$ ?

- d) Lag et nytt program som spør brukeren om en temperatur i antall grader celsius, og så skriver ut den tilsvarende temperaturen.
- e) Bruk programmet ditt til å finne frysepunktet og kokepunktet til vann i Fahrenheit målestokken.

## Oppgave 6 *Jordkloden*



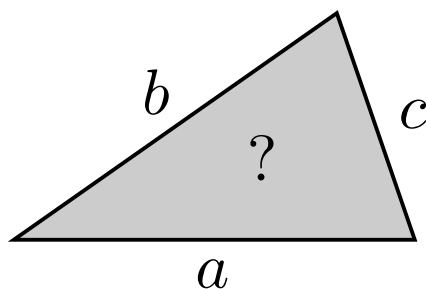
I denne oppgaven skal vi øve på å bruke Python som kalkulator, ved å regne litt på jordkloden. Husk at formelen for volumet av en kule er

$$V = \frac{4}{3}\pi R^3.$$

- a) Jordkloden er tilnærmet en perfekt kule, og har en radius på 6371 km. Lag et kort program som først definerer en variabel `radius`, og deretter regner ut en variabel `volum`. Skriv til slutt ut svaret til brukeren med `print()`-funksjonen. La svaret være i  $\text{km}^3$ .
- b) Endre programmet ditt så svaret istedet skrives ut i antall liter.
- c) Den totale massen til jordkloden er omtrent  $M = 5.972 \cdot 10^{24}$  kg. Regn ut hvor mange kg hver liter av jordkloden veier i gjennomsnitt. Virker svaret ditt rimelig?

### Oppgave 7 *Hérons formel*

Hérons formel er et matematisk uttrykk vi kan bruke for å finne arealet av en trekant, hvis vi kjenner lengdene på de tre sidene av trekanten. Herons formel fungerer for alle trekantar, de trenger ikke for eksempel å være rettvinklet.



For å bruke Herons formel må vi først regne ut et tall  $s$ , som er halve omkretsen til trekanten,

$$s = \frac{a + b + c}{2}.$$

Når vi har funnet  $s$  kan vi regne ut arealet til hele trekanten med formelen:

$$A = \sqrt{s(s-a)(s-b)(s-c)}.$$

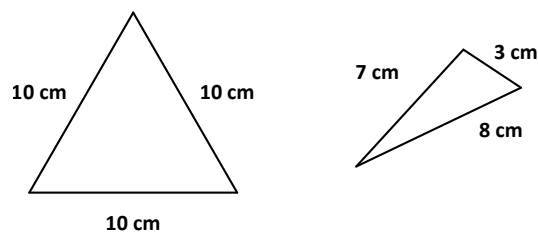
- a) Regn ut arealet av en trekant med sidelengder på 3, 4, og 5 for hånd, ved hjelp av Herons formel.

Vi skal nå skrive et program, som spør brukeren om lengdene på de tre sidene, og så regner ut arealet for oss.

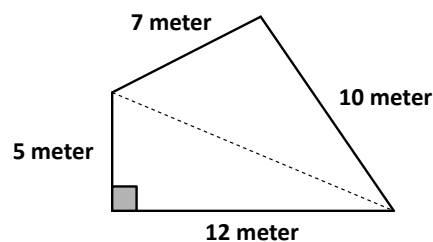
- b) Bruk `float(input(...))` for å spørre brukeren om lengden på de tre sidene av en trekant. Lagre svarene fra brukeren i tre variabler.
- c) Regn så ut variabelen  $s$ , som var halve omkretsen til trekanten.
- d) Regn så ut arealet  $A$  og skriv resultatet tilbake så brukeren kan lese det.

Om du tror du har klart å skrive hele programmet er det på tide å teste det:

- e) Bruk programmet ditt til å sjekke trekanten med sider på 3, 4, og 5. Får du samme svar som du gjorde for hånd?
- f) Bruk programmet til å finne arealet av de to trekantene under.



- g) En firkantet park er formet som figuren under. Klarer du å finne arealet av parken?





## 1.3 Kjemi

### Oppgave 8 Konsentrasjonskalkulator

Når man gjør eksperimenter i kjemien må man ofte lage løsninger med en bestemt molar konsentrasjon.

Si at vi ønsker å lage en løsning av et stoff X, som veier  $M$  g/mol. Vi ønsker å lage en løsning av volum  $V$  mL av en gitt konsentrasjon  $c$  mol/L.

- a) Lag et program som regner ut hvor mange gram av stoff X man må veie ut og løse opp, oppgitt med en nøyaktighet på 0.01 gram.

**Hint:** For å skrive ut en variabel med 3 desimaler bruk print-formatering med `{m:.2f}`.

- b) Test programmet ved å finne mengden bordsalt (NaCl) du trenger for å lage 100 mL løsning med 2.5 mol/L.
- c) Ved 25°C klarer man å løse opptil 35,7 gram NaCl per 100 mL vann, etter dette er løsningen mettet. Øk konsentrasjonen  $c$  i programmet ditt til du er omtrent på denne grensa, hva slags konsentrasjon svarer dette til?

### Oppgave 9 Regne ut pH

pH-verdien til en væske eller løsning er et mål på hvor sur væsken er, det vil si konsentrasjonen av hydrogenioner  $[H^+]$ . Formlene for å regne med pH er gitt ved

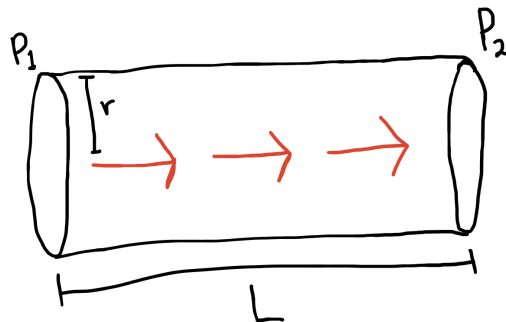
$$\text{pH} = -\log_{10}([H^+]), \quad [H^+] = 10^{-\text{pH}}.$$

- a) Bruk et Python program til å regne ut konsentrasjonen av hydrogenioner for en væske med pH 3.6.
- b) Bruk et Python program til å regne ut pH verdien til en væske med konsentrasjon av hydrogenioner lik  $4.7 \cdot 10^{-5}$  mol/L. **Hint:** for å regne ut 10-logaritmen kan bruke funksjonen `log10` som finnes i `math`, `pylab` og `numpy`

- c) Cola har en pH på ca 2.5, nøytralt vann har en pH på 7. Hvor mange ganger fler hydrogenioner er det i cola sammenlignet med vann?

## 1.4 Fysikk/biologi

### Oppgave 10 Poiseulles lov og blodstrøm



Volumstrømmen i blodårer kan uttrykkes ved hjelp av Poiseulles lov. Poiseulles lov angir volumstrømmen i et sylindereformet rør, volum per tidsenhet, og kan uttrykkes slik:

$$q_V = \frac{\pi r^4 (p_1 - p_2)}{8\eta L} \quad (1)$$

Komponentene i telleren er  $\pi = 3.14$ , radius  $r$  samt trykket i starten og slutten av cylinderen, angitt ved  $p_1$  og  $p_2$ . I nevneren finner vi konstanten 8, viskositeten  $\eta$  (som for blod er 3-4 centipoise, vi kan gi den et gjennomsnitt på 3,5) og lengden av cylinderen  $L$ .

- Lag et program som tar inn alle parameterverdiene fra bruker ved hjelp av `input`, og lagrer disse i hver sin variabel. Husk å konvertere til flyttall ved hjelp av `float`.
- Utvid programmet ditt til å regne ut volumstrømmen ved hjelp av formelen over, og skriv ut en informativ melding om hva væskeflyten blir til bruker.

- c) Finn et sett av egenvalgte parametere. Prøv å doble en og en verdi. Hvilken av parametrene du endrer fører til størst endring i volumstrøm?
- d) Blodtrykket (gitt ved  $p_1$  og  $p_2$ ) varierer typisk mellom 80 og 120 mmHg. Blodårene i kroppen har ulik størrelse. Aorta, hovedpulsåren, har en radius på ca. 10 mm. Denne forgrener seg til større arterier, og videre til mindre arterier (arterioler), som til slutt har en radius som varierer mellom 5 og 100  $\mu\text{m}$ .

Gitt  $p_1 = 100$ ,  $p_2 = 95$  mmHg og lengde  $L = 1$  mm, regn ut volumstrømmen for en arterie med radius

- 10 mm
- 1 mm
- 5  $\mu\text{m}$

## 2 Betingelser

### 2.1 Grunnleggende programmering

#### Oppgave 11 *and* og *or*

I denne oppgaven skal vi se på hvordan *and*- og *or*-operatorene fungerer og hvordan de brukes i betingelser.

- a) Lag en variabel som inneholder tallet 5. Bruk en betingelse til å teste om tallet er mindre enn 10, og print isåfall ut en melding.
- b) Bruk nøkkelordet *and* til å teste om tallet er mindre enn 10 og større enn 2.
- c) Bruk nøkkelordet *or* til å teste om tallet er mindre enn 6 eller større enn 10.

#### Oppgave 12 *and* og *or* 2

Skriv om de følgende programlinjene blir True eller False. Du trenger ikke skriv noe kode i denne oppgaven, men er du usikker på hva svaret er kan du selv prøve å se hva som skjer ved å taste det inn i Python.

```
1 tall = 5 #setter variabelen tall lik 5
2
3 tall == 5
4 tall == 7
5 tall > 8
6 tall > 2
7 tall >= 5
8 tall > 5
9 not True
10 True != False
11
12 True or False
13 False and True
14 True and True
15
16 tall == 5 or tall == 6
```

```

17 tall > 5 and tall == 1
18 tall < 10 and tall > 2
19 tall < 10 or tall > 6
20
21 (tall > 4 or tall == 2) and tall > 6
22 (tall == 5 or tall > 4) or True
23 (tall == 5 and tall > 4) or (tall < 4 or tall > 1)
24
25 not (tall == 5)
26 tall != 4
27 not (tall != 10)
28
29 (not (tall < 5) and (tall > 1)) or not tall != 5

```

## 2.2 Matematikk

### Oppgave 13 *Absoluttverdi*

Et reelt tall består av et fortegn og en tallverdi, kalt *absoluttverdi*. Når vi finner absoluttverdien til et tall «fjerner vi fortegnet». Det betyr at absoluttverdien til et tall alltid er positiv. Absoluttverdien til et tall  $a$  skrives  $|a|$  og er definert som:

$$|a| = \begin{cases} a, & \text{hvis } a \geq 0 \\ -a, & \text{hvis } a < 0 \end{cases} \quad (2)$$

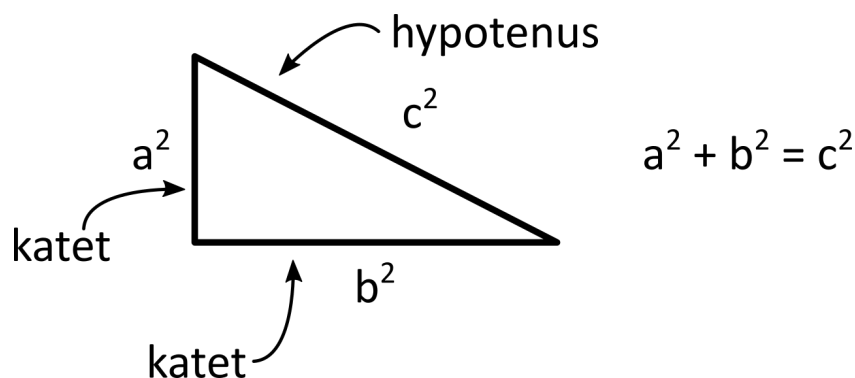
- Lag et program som ber brukeren om et tall ved hjelp av `input` og `float`, og skriver ut absoluttverdien av tallet
- Lag et program som ber brukeren om to tall og skriver ut hvilket av tallene som har høyest absoluttverdi

### Oppgave 14 *Sjekk alder*

En film på kino har aldersgrense 15 år. Vi skal lage et program som interagerer med brukeren, og finner ut om de kan se filmen eller ikke.

- a) Lag et program som printer ut en hilsen til brukeren, og spør hvor gamle de er. Lagre alderen i variabelen *alder*.
- b) Lag en test som sjekker om brukeren er 15 år eller eldre. Skriv ut passende svar avhengig av om de er gamle nok til å se filmen eller ikke.
- c) Utvid programmet ditt til å regne ut hvor mange år det er til brukeren kan se filmen.
- d) La programmet ditt gjøre et unntak for de som er 12 år eller eldre dersom de har med seg en voksen.

### Oppgave 15 *Pytagoras*



- a) Skriv et program som spør brukeren om lengden på de to katetene i en rettvinklet trekant, og så skriver ut lengden på hypotenusen
- b) Utvid programmet ditt, slik at det først spør brukeren om de vil finne en katet eller en hypotenus. Avhengig av hva brukeren svarer må du deretter spørre om lengden på de to kjente sidene og regne deg frem til den ukjente siden.

### Oppgave 16 *Spisse og butte trekanter*

En trekant kalles *spiss* dersom alle de tre vinklene til trekanten er spisse (altså  $< 90^\circ$ ), tilsvarende kalles trekanten *butt* dersom én av de tre vinkelene er butt (altså  $> 90^\circ$ ). En trekant vil alltid enten være spiss, rettvinklet, eller butt.

Dersom vi kaller de tre sidelengdene  $a$ ,  $b$  og  $c$  og lar  $c$  være den lengste av disse kan vi vise at

$$a^2 + b^2 \begin{cases} > c^2 & \text{for spiss trekant,} \\ = c^2 & \text{for rettvinklet trekant,} \\ < c^2 & \text{for butt trekant.} \end{cases} \quad (3)$$

For rettvinklede trekanter er dette bare Pytagoras' læresetning. Men vi ser altså at likheten i Pytagoras' endres til ulikheter for spisse og butte trekanten.

Du skal nå lage et program der brukeren oppgir de tre sidelengdene i trekanten  $a$ ,  $b$  og  $c$  og programmet skal si om trekanten er spiss, butt eller rettvinklet.

a) Forklar skjelettkoden under og fyll inn de bitene som mangler

```
1 print("Skriv inn de tre sidelengdene a, b og c der
   c skal være den lengste.")
2 a = float(input("Sidelengde a: "))
3 b = float(input("Sidelengde b: "))
4 c = float(input("Sidelengde c: "))
5
6 if c < a or c < b:
7     raise ValueError("c må være den lengste
   sidelengden.")
8
9 if ...:
10     print(...)
11
12 elif ...:
13     print(...)
14
15 elif ...:
16     print(...)
```

b) Vi gir nå sidelengdene til fem ulike trekanter. Hvilke to er spisse, hvilke to er butte og hvilken er rettvinklet?

- 4, 6, 8
- 3, 5, 5

- 5, 12, 13
- 6, 6, 8
- 9, 13, 17

### Oppgave 17 *Hvilken skala?*

Et hus kan ha lengde 1 314 cm – eller, muligens på en mer hensiktsmessig skala, omtrent 13 meter. En celle kan være 0.000054 m lang – altså 54  $\mu\text{m}$ . En fotballbane kan ha et areal på 6443399361  $\text{mm}^2$  – eller ca 6405  $\text{m}^2$ .

I denne oppgaven skal vi, litt trinn for trinn, bygge opp et program for å konvertere *lengdemåleneheter* til en hensiktsmessig skala. Deretter vil vi gjerne utfordre dere på hvordan dere vil tenke for å utvikle et lignende program for å konvertere areal eller volum på lignende måte.

Vi anser at en målenhet er «hensiktsmessig» å bruke dersom målet kan oppgis med et tall større enn 0 foran kommaet, og at vi bruker størst mulig skala der dette er mulig. For å sjekke om man får et positivt tall foran kommaet (dvs. punktumet i Python) kan man konvertere tallet til et heltall ved hjelp av `int` og så sjekke om dette blir større enn 0.

- Lag et Python-program som spør en bruker om hvor langt noe er. Lagre dette i en variabel. Husk å konvertere lengden til et desimaltall (ved hjelp av `float()`).
- Spør deretter brukeren om hvilken målenhet dette er i. Her bør dere opplyse om hvilke mulige enheter det kan være – start med «m, cm eller mm».
- Bruk betingelser, basert på målenheten brukeren gir, til å konvertere lengden brukeren har oppgitt til meter. Skriv ut svaret midlertidig og prøv noen ganger med regning for å sjekke at du har programmert dette riktig.
- Regn ut hva lengden også blir i de andre enhetene (ut fra lengden i meter), og lagre disse i nye variable.



- e) Bruk en betingelse for å først sjekke om det er hensiktsmessig å oppgi enheten i den største enheten (altså m). Hvis ja, skriv ut hva det blir i meter – hvis ikke, sjekk om det er hensiktsmessig å oppgi den i den neste (cm), osv. Hvis det hverken er hensiktsmessig å oppgi målenheten i m eller cm, skriv ut svaret i den siste mulige (mm).
  - f) Skriv ut svaret på en fin måte. Bestem dere for antall desimaler dere vil oppgi i svaret. Dere kan formatere tallet ved å skrive et kolon, et punktum, antall desimaler og *f* etter variabelen i utskriften, altså f.eks. `print(f"... {lengde_meter:.2f} ...")`.
  - g) Utvid programmet deres til å også akseptere km og  $\mu\text{m}$  som mulige enheter.
  - h) Finn på fem objekter med lengde, i forskjellige størrelsesordener, og prøv om programmet gir forventet resultat for disse.
  - i) Nå har dere skrevet et program som kan konvertere ulike lengder til en skala som (muligens) er mer hensiktsmessig for det man vil måle. Hvordan ville dere utviklet et tilsvarende program for konvertering av areal? Av volum? Hva ville vært likt som i dette programmet, og hva ville vært annerledes? Hva ville vært de matematiske utfordringene?
- Diskuter spørsmålene over med en kollega eller i en gruppe. Det er også mulig å programmere dette for å utfordre seg selv – eller kanskje elevene?

### Oppgave 18 *Finn riktige typer fliser*

Vi skal legge fliser på kjøkkenet, og har en flate på  $75 \times 350 \text{ cm}^2$  som skal dekkes. Fra en butikk har vi funnet et utvalg av fliser som vi synes er fine, og nå vil vi sjekke om vi kan bruke noen av disse slik at vi får hele fliser hele veien – altså om noen av de har mål som passer akkurat både i høyden og bredden. For enkelthets skyld antar vi at flisene legges helt tett, det vil si at vi ikke trenger å beregne plass til fugemasse imellom.

Flisene vi har funnet har følgende mål:

1.  $25 \times 15 \text{ cm}^2$
2.  $25 \times 25 \text{ cm}^2$
3.  $23 \times 10 \text{ cm}^2$

4.  $30 \times 60 \text{ cm}^2$

5.  $9 \times 9 \text{ cm}^2$

6.  $6 \times 12 \text{ cm}^2$

For å sjekke om antall fliser vil passe én vei kan vi bruke *modulo*-operatoren, som vil gi oss *resten* i et delestykke. I Python bruker man % til dette. For eksempel vil

```
1 print(5 % 2)
```

gi oss

```
1
```

siden 5 går 2 ganger opp i 2, med rest 1. Vi kan sjekke om antall fliser passer akkurat ved å sjekke at vi får rest 0.

- a) Lagre alle målene i variabler.
- b) Bruk modulooperatoren for å sjekke om flis nr. 1 vil passe akkurat i *høyden*. Skriv programmet ditt slik at den både skriver ut en beskjed om flisen passer, og om den ikke gjør det.
- c) Bruk modulooperatoren og den logiske operatoren **and** for å sjekke om flis nr. 1 også vil passe akkurat i bredden.
- d) Gjør det samme for flis nummer 2 (her kan du kopiere koden din fra a og b, og så endre variabelnavnene).
- e) Dersom antall fliser passer akkurat, bruk heltallsdivisjon (`//`) for å finne ut hvor mange fliser vi trenger totalt, og skriv ut resultatet. Legg inn dette både for flis 1 og flis 2.
- f) Endelig skal vi sjekke alle flisene: Gjenta (kopier) koden over så mange ganger som nødvendig, og sjekk for hver flis om den kommer til å passe eller ikke.

- g) Vi vurderer noen andre mål. Hva skjer om man endrer målene på flaten til  $90 \times 360 \text{ cm}^2$ ? Eller  $81 \times 360 \text{ cm}^2$ ? Vil noen av fliskombinasjonene passe da? Er det flere som kan passe?

## 2.3 Naturfag

### Oppgave 19 *Vann og sjokolade ved forskjellige temperaturer.*

Smeltepunktet til et stoff markerer den temperaturen som gjør at stoffet endrer fasetilstand mellom fast og flytende form. Vann har smeltepunkt ved 0 grader celsius, og sjokolade har smeltepunkt ved ca. 40 grader celsius.

Lag et program som spør om en temperatur, og lag **if**-tester for å sjekke hvilken fasetilstand vann og sjokolade er i ved denne temperaturen. Print til slutt ut en setning som nevner temperaturen og tilhørende fasetilstand for stoffene.

### Oppgave 20 *Beskrive tilstanden til kjemiske stoff*

En venn av deg har lagd et program som forteller deg om vann fryser eller ikke basert på temperatur. Men det finnes jo flere stoffer enn vann, og du har lyst til å lage et program som kan fortelle deg om de er fast form, væske eller gass.

Programmet til vennen din ser du her:

```
1 kokepunkt = 100 # grader Celcius
2 frysepunkt = 0 # grader Celcius
3 stoff = 'vann'
4
5 temperatur = float(input(f'Hvilken temperatur har {
    stoff}?'))
6
7 if temperatur < frysepunkt:
8     print('Det er et fast stoff')
9 elif temperatur < kokepunkt:
10    print('Det er en væske')
11 else:
12    print('Det er en gass')
```

- a) Utvid programmet til at du bruker `input` funksjonen til å spørre brukeren om hvilket stoff den vil vite fasen til.
- b) Bruk en `if` og en betingelse til å sjekke om brukeren skriver vann. Hvis brukeren skrev vann skal kokepunkt variabelen settes til 100 og frysepunkt variabelen settes til 0. Hvis brukeren skrev noe annet enn 'vann' skal 'Jeg vet ikke om dette stoffet' skrives ut til brukeren. NB: Hvis brukeren skriver inn noe annet enn vann så vil denne beskjen printes ut, og så vil vi få en feilmelding!
- c) Nå må vi passe på at programmet avsluttes uten å krasje dersom brukeren skriver inn noe annet enn vann. For å gjøre det trenger vi `exit` funksjonen fra `sys` biblioteket. Start derfor programmet ditt med å hente denne funksjonen ved å skrive `from sys import exit`.
- d) Modifiser `else`-blokken til programmet ditt, slik at du kaller på funksjonen `exit()` etter at 'Jeg vet ikke om dette stoffet' skrives ut til brukeren. Test programmet ditt, hva skjer om du skriver inn et stoff det ikke kjenner til (f.eks. saltvann)
- e) Legg til en mulighet for å skrive *håndsprit*. Håndsprit fryser på ca  $-89$  grader og koker på ca  $83$  grader.
- f) Utvid programmet til å støtte kvikksølv og luft. Kvikksølv fryser på ca  $-39$  grader og koker på ca  $357$  grader, luft fryser på ca  $-210$  grader og koker på ca  $-196$  grader.

## 3 Løkker

### 3.1 Grunnleggende programmering

#### Oppgave 21 *For-løkker for hånd*

For hver løkke, gå igjennom for hånd og forutsi hva som skrives ut. Etterpå kan du kjøre løkkene og se hva om du hadde rett:

a)

```
1 for tall in range(1, 5):  
2     print(tall)
```

b)

```
1 sum = 0  
2 for tall in range(1, 5):  
3     sum += tall  
4     print(sum)
```

c)

```
1 produkt = 0  
2 for tall in range(1, 5):  
3     produkt *= tall  
4     print(produkt)
```

d)

```
1 produkt = 1  
2 for tall in range(1, 5):  
3     produkt *= tall  
4     print(produkt)
```

e)

```
1 x = 1
2 for _ in range(10):
3     x *= 2
4     print(x)
```

## 3.2 Matematikk

### Oppgave 22 *Renter*

Bank 1 gir fast 3 prosent rente på sin sparekonto. Bank 2, derimot, gir 3,3 prosent rente de første fem årene før de skifter til 2,8 prosent rente. Du skal sette 10000, – i en bank i morgen. I denne oppgaven skal du bruke **for**-løkker til å simulere hva som skjer med pengene i de ulike bankene.

- a) Hvilken bank er best å bruke hvis du skal spare i ti år?
- b) Hvor lenge må du ha pengene i bank 1 for at det skal lønne seg fremfor bank 2?

Naboen din bestemmer seg for å heller sette inn 1000, – hver januar, istedenfor å sette inn en engangssum slik som du gjør.

- c) (Bonusoppgave) Hvilken bank er det best for naboen din å bruke hvis han skal spare i ti år?
- d) (Bonusoppgave) Hvor lenge må han ha pengene i bank1 for at det skal lønne seg fremfor bank 2?

### Oppgave 23 *Kjøpe telefon på kreditt*

Hallgeir har veldig lyst på en ny telefon som koster 2999 kroner. Problemet er bare at han har brukt opp sparepengene sine på andre ting. For å få kjøpt telefonen skaffer Hallgeir et kredittkort som har 30% rente. Etter å ha kjøpt telefonen går det tre år før Hallgeir betaler tilbake kreditten. I denne oppgaven skal vi undersøke hvor mye Hallgeir blir nødt til å betale da.

- a) Opprett variablene renter, originalpris og antall\_år og gi med verdiene 30, 2999 og 3
- b) Regn ut vekstfaktoren til renta og lagre den i en variabel rentevkstfaktor
- c) Opprett en variabel, lån. Denne variabelen skal holde orden på hvor stort lånet til Hallgeir er. Til å begynne med er lånet like stort som prisen på telefonen. Sett altså lån-variabelen til å ha verdien 2999
- d) Bruk en for-løkke til å simulere hvordan lånet vokser for hvert år. Hint: for hvert år skal lånet ganges med vekstfaktoren du regnet ut i oppgave b)
- e) Oppdater programmet ditt til å skrive ut størrelsen på lånet for hvert år. Hvor mye skylder Hallgeir etter tre år?
- f) Hvor mye ekstra kostet telefonen i forhold til originalprisen?
- g) Gå inn på <https://kredittkort.com/> og se hvilket kort som kommer øverst og noter deg renta. Endre renta i programmet ditt til å matche denne renta. Gå inn på <https://www.prisjakt.no/category.php?k=103> og se hvilken telefon som er mest populær. Endre orginalprisen i programmet ditt til å matche prisen til den telefonen. Kjør programmet nå. Hva ville denne telefonen og dette kredittkortet kostet Hallgeir?

#### Oppgave 24 Gangetabell-quiz

- a) Lag et program som trekker to tilfeldige tall og spør brukeren om hva produktet av dem er. (Hint: legg til linjen `import random` på første linje. Da vil `tilfeldig_tall = random.randint(1,10)` være et tilfeldig tall mellom 1 og 10.)
- b) Modifiser programmet ditt slik at du sjekker om svaret var riktig, og om så var tilfelle så får brukeren et poeng. (Hint: lag en variabel poeng = 0 som du øker med en hvis svaret er rett.)
- c) Bruk det du har programmert så langt sammen med en while-løkke. La programmet spørre brukeren flere ganger om hva produktet av to tall

er, og hvis svaret er riktig så får brukeren et poeng. Programmet skal slutte når brukeren får mindre enn - 10 poeng eller mer enn 10 poeng.

- d) Skriv ut en melding helt til slutt om brukeren vant (mer enn 10 poeng) eller tapte (mindre enn -10 poeng).

### Oppgave 25 *Tverrrsum*

Tverrrsummen av et tall er tallet du får dersom du plusser alle sifrene i tallet med hverandre.

- a) Hvor mange 3-sifrede tall finnes det som har tverrrsum 5? Løs for hånd.
- b) Skriv et program som finner alle 3-sifrede tall med tverrrsum 4. Start med å skrive en løkke for alle tall fra 1 (hvorfor fra 1?) til 9; i denne løkken lager du enda en løkke; og i denne løkken trenger du enda en. I den innerste løkken trenger du en betingelse, og til slutt en **print**.
- c) Legg inn en teller i programmet ditt, som først er 0, og deretter øker hver gang du finner et tall med tverrrsum 4. Hvor mange tall er det tilsammen? Stemmer det med det du fikk når du regnet for hånd?
- d) Tror du det er flest 3-sifrede tall med tverrrsum 4, eller flest 4-sifrede tall med tverrrsum 3 – eller er det like mange? Utvid programmet ditt til å telle begge deler, og sjekk om du hadde rett.

### Oppgave 26 *Fibonacci rekken*

Fibonacci rekken er en kjent tallfølge som naturlig oppstår mange steder i naturen. Tallfølgen går som følger:

$$1, 1, 2, 3, 5, 8, 13, 21, \dots \quad (4)$$

Vi kan skrive Fibonacci rekken som en rekursiv tallfølge etter denne formelen:

$$a_n = a_{n-1} + a_{n-2}, \quad (5)$$

hvor  $a_0 = 1$  og  $a_1 = 1$ .

En interessant egenskap ved Fibonacci rekken er at forholdet mellom to etterfølgende tall i følgen går mot det gyldne snitt,  $\phi = \frac{1+\sqrt{5}}{2} \approx 1.62$ . Det vil si at



$$\frac{a_n}{a_{n-1}} \rightarrow \phi. \quad (6)$$

- a) Du skal nå lage et program som skriver ut de første 10 tallene i Fibonacci rekka. Start med å opprette en variabel `forrige_tall = 1` og en variabel `fibonacci_tall = 1`.
- b) Skriv så ut `forrige_tall` og `fibonacci_tall` til brukeren av programmet.
- c) Bruk en **for**-løkke som repeteres 8 ganger (10 tall - 2 start-tall) som skriver ut de resterende 8 tallene i Fibonacci-rekka. HINT: Her kan det være lurt å opprette det nye Fibonacci tallet i en variabel `nytt_fibonacci_tall` før du oppdaterer verdien til `forrige_tall` og `fibonacci_tall`.
- d) Endre programmet slik at du bruker input til å be brukeren om hvor mange Fibonacci tall du skal skrive ut til skjermen.
- e) Oppdater programmet slik at du og skriver ut forholdet mellom `forrige_tall` og `fibonacci_tall`. Blir dette forholdet ca lik 1.62?
- f) Endre start-tallene fra  $a_0 = 1$  og  $a_1 = 1$  til noe annet (f.eks  $a_0 = 2$  og  $a_1 = 1$ ), hvordan endrer det oppførselen til rekka?

### 3.3 Naturfag

#### Oppgave 27 *Finne komplementær DNA-streng*

De to trådene som utgjør DNA-dobbelspiralen består av deoksyribose, en fosfatgruppe og de fire nitrogenbasene A, T, C og G. En base fra hver tråd er forbundet med en hydrogenbinding, og kombinasjonene er enten A og T eller C og G.

- a) Lag en DNA-streng på 50 baser med tilfeldig rekkefølge av de 50 basene.
- b) Lag en funksjon som kan ta inn en hvilken som helst DNA-streng og

generere den komplementære tråden.

### **Oppgave 28** *Vektvekst for selunger*

Grøndlandsseler dier ungene sine intensivt de 12 første levedagene. Melken er veldig rik på fett, og selungene legger på seg ca. 2.2 kilo om dagen denne perioden. Nyfødt veier selungene 11 kg.

- a) Bruk en løkke til å finne vekten for selungen de 12 første levedagene og legge de i en liste. Lag også en liste med dagene, 1-12. Print listene.
- b) Finn fra listen over vekt
  - vekten til selen på dag 12
  - vektene for dagene 5-7
- c) Lag et plot med vekt mot dager. Gi aksene og plottet et navn.

## 4 Funksjoner

### 4.1 Grunnleggende programmering

#### Oppgave 29 *Enkle funksjoner*

- a) Lag en funksjon `pluss(a,b)` som tar inn to tall  $a$  og  $b$  og returnerer summen av dem.
- b) Lag en funksjon `minus(a,b)` som tar inn to tall  $a$  og  $b$  og returnerer differansen av dem.
- c) Lag en funksjon `kalkulator(operasjon,a,b)` som tar inn operasjon som enter er "pluss" eller "minus" (i form av en tekststring), og to tall, og regner ut resultatet.

### 4.2 Matematikk

#### Oppgave 30 *Faktorisering*

Alle positive heltall er enten primtall, eller kan faktorerises til et produkt av flere primtall, som i

$$15 = 3 \cdot 5$$

$$20 = 2 \cdot 2 \cdot 5$$

Her skal vi lage et program som finner faktoriseringen av et positivt heltall for oss.

- a) *Modulo*-operasjonen forteller oss om et tall er delelig med et annet tall eller ikke. Denne gir oss resten ved en divisjon, og hvis  $a \bmod b$  (syntaks `a % b` i Python) er lik 0, er  $a$  delelig på  $b$ . Skriv en funksjon `delelig_paa` som tar inn to tall  $a$  og  $b$ , og returnerer `True` hvis  $a$  er delelig på  $b$ , `False` ellers.
- b) Skriv en funksjon `finn_neste_faktor` som tar inn et heltall og returnerer det minste tallet (som er større enn 1) dette er delelig på. For

eksempel vil man at `finn_neste_faktor(20)` skal returnere 2.

- c) Skriv en funksjon `skriv_ut_faktorisering` som tar inn et heltall og sjekker om det er et primtall eller ikke – og hvis ikke, skriver ut primtallsfaktoriseringen av dette.
- d) Hvorfor kan man ikke bruke programmet du skrev over til å finne alle primtall som finnes? Hva skjer hvis du gir større og større tall som input?

## 4.3 Naturfag

### Oppgave 31 *Elektrisitet: Strøm – spenning – resistans*

Viktige egenskaper i en elektrisk krets er spenningen  $U$ , målt i volt, strømmen  $I$ , målt i ampere, og motstanden/resistansen  $R$ , målt i ohm ( $\Omega$ ). Forholdet mellom disse enhetene er gitt ved *Ohms lov*:  $U = R \cdot I$

- a) Lag tre funksjoner som regner ut strøm, spenning og resistansen i kretsen gitt at man vet de to andre størrelsene.
- b) Bruke funksjonene dine til å regne ut:
  - Spenningen når strømmen er 10 A og motstanden 1.7  $\Omega$
  - Resistansen når spenningen er 230 V og strømmen er 20 A
  - Strømmen når spennignen er 5 V og motstanden er 200  $\Omega$
- c) Når flere motstander er koblet i seriekobling, blir den totale motstanden lik summen av alle motstandene i serien.  $R_{tot} = R_1 + R_2 + \dots + R_N$ . Lag en funksjon `seriekobling` som tar inn en liste med motstander og returnerer den totale motstanden.
- d) Dersom spenningen er 5 V, og motstandene på 10, 5, 2 og 11  $\Omega$  er koblet i serie, hva blir strømmen da?
- e) **Utfordring:** Når motstander er koblet i parallell, er sammenhengen mellom dem  $\frac{1}{R_{tot}} = \frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_N}$ . Lag en funksjon `parallellkobling` som regner ut den totale motstanden i kretsen. Hva blir strømmen nå, dersom motstandene og spenningen er den samme?

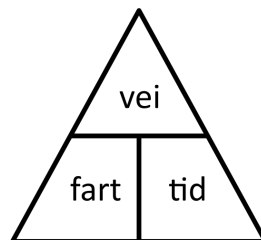
## 4.4 Fysikk

### Oppgave 32 *Vei-fart-tid-kalkulator*

En viktig formel for å beskrive bevegelse er vei-fart-tid formelen,

$$\text{vei} = \text{tid} \cdot \text{fart}$$

Egentlig er det tre formler i ett, for vi kan stokke om på den avhengig av hva vi ønsker å regne ut. Av og til tegnes vei-fart-tid formelen som en pyramide:



- a) Lag en funksjon som bruker vei-tid-fart-formelen gitt over, kall denne `vei(fart, tid)`. Vi vil at fart skal oppgis i kilometer i timen, vei i antall kilometer og tid i antall minutter – husk å regne om tiden fra antall timer til antall minutter.
- b) Skriv opp uttrykket for å regne ut fart, gitt vei og tid. Lag en funksjon som bruker formelen, kall denne `fart(vei, tid)`.
- c) Skriv opp uttrykket for å regne ut tid, gitt vei og fart. Lag en funksjon som bruker formelen, kall denne `tid(vei, fart)`.
- d) Du skal nå lage en vei-fart-tid-kalkulator. Når dette programmet kjører skal det først spørre brukeren om de ønsker å regne ut vei, fart, eller tid. Avhengig av svaret skal så programmet spørre om de to andre størrelsene, og så regne ut og skrive ut svaret.

### Oppgave 33 *Hvor mye koster morgendusjen din?*

For å varme opp vann trenger vi energi, og i Norge kommer den energien vanligvis fra strøm. Enheten som brukes for å måle hvor mye strøm vi har brukt er gjerne kilowattimer (kWh), som representerer at du har brukt 1000 Watt i en time.

Til vanlig måler vi energiforbruk i Joule (J), så derfor kan det være nyttig å ha en funksjon som oversetter energi fra kWh til J og motsatt. Det kan vi gjøre med denne formelen

$$[\text{kWh}] = 3.6[\text{MJ}], \quad (7)$$

hvor MJ er megajoule, eller  $10^6$  J.

Når vi varmer opp vann, så bruker vi 4.2 J per liter (L) vann. For å gjøre enhetsomregning lettere er det derfor vanlig å snakke om enheten *kalorier* (cal). En kalori representerer mengden energi som skal til for å varme opp en liter vann en grad Celcius ( $^{\circ}\text{C}$ ). Altså er en kalori gitt ved formelen

$$[\text{cal}] = 4.2[\text{kJ}], \quad (8)$$

hvor kJ er kilojoule, eller 1000 J.

- a) Lag en funksjon `kWh_til_joule(energi_i_kWh)` som oversetter en energimengde oppgitt i kWh til en energimengde oppgitt i Joule.
- b) Lag en funksjon `joule_til_kWh(energi_i_J)` som oversetter en energimengde oppgitt i Joule til en energimengde oppgitt i kWh.
- c) Lag en funksjon `kalori_til_joule(energi_i_cal)` som oversetter en energimengde oppgitt i kalorier til en energimengde oppgitt i Joule.
- d) Lag en funksjon `kalori_til_kWh(energi_i_cal)` som oversetter en energimengde oppgitt i kalorier til en energimengde oppgitt i kWh. (Tips: Bruk `joule_til_kWh`- og `kalori_til_joule`-funksjonene du allerede har laget).
- e) Når vi dusjer bruker vi fort 60 L vann, og dette vannet blir gjerne varmet opp 35 grader. Bruk disse tallene for å estimere hvor mye energi (i kWh) du bruker hver gang du dusjer.

f) En tommelfingerregel er at hver kWh med energi vi kjøper koster ca 1 krone. Hvor mye penger bruker du da på å dusje hvis du dusjer hver dag i et år?

## 5 Plotting

### 5.1 Matematikk

#### Oppgave 34 *Innhegning*

Du har 100 meter gjerde og vil lage en rektangulær innhegning. Vi skal se på hvor stort areal denne innhegningen vil ha.

- a) Tegn opp et rektangel på papir. Hvis du sier at den ene siden er  $x$  meter lang, hvor lang blir da de tre andre sidene i rektangelet? Skriv det opp på arket.
- b) Skriv ut uttrykket for arealet til hele innhegningen.
- c) Hvor stor kan  $x$  maksimalt være?

Du skal nå lage et Python-program som plotter arealet av innhegningen som en funksjon av  $x$ .

- d) Fyll inn i skjelettkoden under for å lage programmet:

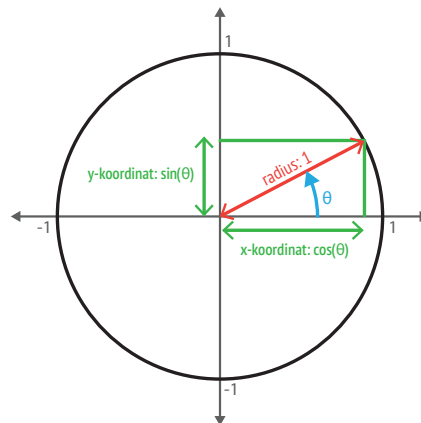
```
1  from pylab import *
2
3  x = arange(0, ..., 0.1)
4  A = ...
5
6  plot(x, A)
7  xlabel('x')
8  ylabel('Areal')
9  show()
```

- e) Kjør programmet og se på figuren som tegnes. Hvilket valg av  $x$  er det som gir størst mulig areal? Hva slags innhegning er det vi ender opp med å lage?

#### Oppgave 35 *Sirkelparameterisering*



I denne oppgaven skal vi bruke Python til å tegne en sirkel med radius lik 1 som er sentrert i origo. En slik sirkel er gitt ved parametriseringen  $x(\theta) = \cos(\theta)$  og  $y(\theta) = \sin(\theta)$ . Hvis du lurer på hvorfor sirkelen har denne parametriseringen er lurt å legge merke til at det vi ønsker å tegne rett og slett er en enhetssirkel. Og hvis man har vinkelen  $\theta$  så er  $x$ - og  $y$ -koordinaten til punktet med den vinkelen gitt ved  $x = \cos(\theta)$  og  $y = \sin(\theta)$

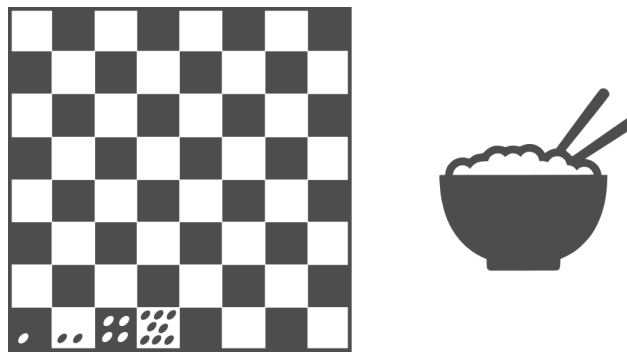


- Bruk `linspace` til å opprette en *array*, `theta`, som skal inneholde 100 tall fra 0 til  $2\pi$
- Opprett en variabel `x` som inneholder  $x$ -koordinatene til punktene på sirkelen ( $x$  koordinatene er gitt ved  $\cos(\theta)$ )
- Opprett en variabel `y` som inneholder de korresponderende  $y$ -koordinatene til punktene på sirkelen (altså  $\sin(\theta)$ ).
- Tegn sirkelen med `plot`-kommandoen (Hint: husk `show`).
- Se på sirkelen, er det noe som ikke stemmer?
- For å få sirkelen til å være sirkulær, ikke elliptisk, så må en centimeter på  $x$ -aksen tilsvare en centimeter på  $y$ -aksen. For å få til dette kan vi skrive `axis("equal")`. Det tvinger aksene til å ha samme skalering. Legg til denne kommandoen i programmet ditt. Kjør koden og se på plottet, har vi en sirkel nå?

### Oppgave 36 *Sjakk og riskornvekst*

Det finnes en nesten tusen år gammel legende om oppfinnelsen av sjakk som går slik: En veldig smart mann finner opp spillet sjakk og viser det til keiseren sin. Keiseren blir så imponert at han erklærer at oppfinneren kan velge sin egen belønning. Oppfinneren svarer at han er en ydmyk mann og ønsker kun ris. Og siden det er sjakk han blir belønnet for vil han ha et riskorn for den første ruta i brettet, to for den andre, fire for den tredje og så videre. Han ønsker altså at mengden ris skal dobles for hver rute i brettet.

Kongen synes dette er en beskjeden belønning og aksepterer den på stedet. Men når han forteller det til sin kasserer får han beskjed om at hele keiserdømmet vil gå konkurs!



- Bruk en for-løkke til å finne ut hvor mange riskorn oppfinneren ba om. Er du enig med kassererens fortvilelse?
- Det er omtrent 50 000 riskorn i et kilo med ris. Bruk en **while**-løkke til å finne ut hvor mange ruter man må belønne oppfinneren for for at han skal få minst et kilo med ris.
- Lag et plot over veksten av riskorn. La  $x$ -aksen være antall sjakkruiter og  $y$ -aksen være antall riskorn. Gjør plottet pent ved å gi navn til aksene med `xlabel(...)/ylabel(...)`, legg på en tittel med `title(...)` og et rutenett med `grid()`.

## 5.2 Naturfag

### Oppgave 37 Vekstfaktor

I en petriskål lever en bakteriekultur som består av 1000 bakterier som former seg raskt nok til at mengden bakterier øker med 50% hver time. I denne opppgaven skal vi bruke plotting og python til å modellere og visualiserer veksten i bakteriekulturen over 10 timer

- a) Opprett en variabel, vekstfaktor, som har vekstfaktoren tilhørende 50%
- b) Bruk arange fra pylab til å opprette en array, timer, med tallene fra 1 til 10

Nå vil vi regne ut hvor mange bakterier som er i petri-skålen hver time. For å gjøre det, må vi først opprette en tom array og bruke en løkke til å "fylle" inn antall bakterie hver time. Men først, hva er egentlig en tom array? Vel, vi kan jo bruke en array hvor alle elementene er lik 0. Dette er det en funksjon for i pylab fra før av, og den heter zeros. Hvis vi skriver `ti_nuller = zeros(10)`, vil vi få en variabel `ti_nuller` som består av ti element som alle er lik null.

- c) Bruk zeros fra pylab til å opprette en tom array, bakteriemengde med 10 elementer. Husk å importere zeros først!

Det neste steget er å endre verdiene til hvert element i arrayen vårt. Dette kan vi gjøre med *indeksering*. En array består av mange tall som er etter hverandre, og hvis du vil ha ut et tall fra en array bruker vi klammeparanteser. Hvis vi har en array-variabel, `x`, kan rett og slett skrive `x[0]` for å hente ut det første elementet i arrayen. Tilsvarende kan vi skrive `x[1]` for å hente ut det andre elementet i arrayen, osv. Under har vi et eksempel

```
1 from pylab import arange
2
3 tallrekke = arange(10)*2
4 første_tall = tallrekke[0]
5 femte_tall = tallrekke[4]
6
7 print(første_tall)
8 print(femte_tall)
```

```
0
8
```

Tilsvarende, kan vi starte med en tallrekke, og endre enkeltelement! Se eksempelet under.

```
1 from pylab import arange
2
3 tallrekke = arange(4)*2
4 print(tallrekke)
5
6 tallrekke[0] = -1
7 tallrekke[2] = 0
8
9 print(tallrekke)
```

```
[0 2 4 6]
[-1 2 0 6]
```

Vi ser her at vi bruker `tallrekke[i] = x` for å sette verdien til element nummer  $i + 1$  i arrayet lik  $x$ .

- d) Bruk array-indeksering (klammeparanteser) til å sette det første elementet (element nummer 0) i bakteriemengde til 1000 som er startverdien.
- e) Bruk en løkke `for time in range(1,10)` til å løkke igjennom alle timene fra 1 til 9
- f) Bruk arrayindeksering til å sette bakteriemengden for tid `time` til å være lik vekstfart multiplisert med bakteriemengden for tid `time-1`. **Hint:** `bakteriemengde[time-1]`
- g) Bruk `plot` og `show` funksjonene fra `pylab` til å plote bakteriemengde på y-aksen og timer på x-aksen
- h) Bruk `xlabel` og `ylabel` funksjonene fra `pylab` til å legge merkelapper på x og y-aksen

- i) Refleksjonsoppgave: Hva vil skje med bakteriene etterhvert som tiden går? Hva synes du om denne modellen? Er det noen svakheter ved en slik modell?