

# Bonus-oppgaver

Her er et knippe bonusoppgaver for de som vil ha litt mer å bryne seg på. Disse oppgavene er ikke sortert på nivå, men etter fag. Samlingen er fin som ekstra programmeringstrening eller som inspirasjon til realfagsrelevante Python-oppgaver.

## 1 Matematikk

### Oppgave 1 *Annengradsfunksjon*

En annengradsfunksjon er definert slik:

$$f(x) = x^2 + 0.3x - 1$$

- a) Definer funksjonen som en Python-funksjon,  $f(x)$ , som tar inn en  $x$ -verdi og returnerer tilhørende  $y$ -verdi.
- b) Test funksjonen din ved å kalle på den med følgende  $x$  verdier og skriv ut resultatet til terminalen:
  - 1.  $x = 0$
  - 2.  $x = 1$
  - 3.  $x = -0.4$

### Oppgave 2 *Arealet av en trekant*

Arealet av et rektangel er gitt ved:

$$\text{lengde} \times \text{høyde}$$

- a) Lag en funksjon, `firkantareal`, som tar inn to tall, lengde og høyde og returnerer arealet av et rektangel med tilsvarende lengde og høyde
- b) Bruk funksjonen til å finne arealet av et rektangel med høyde 31 og lengde 43

Arealet av en trekant er halvparten av arealet til et rektangel som har samme grunnlinje og høyde:

$$\frac{\text{lengde} \times \text{høyde}}{2}$$

I denne oppgaven skal du bruke funksjonen, `firkantareal`, du lagde i oppgave a) til å regne ut trekantareal i stedet

- c) Lag en funksjon, `firkantareal`, som tar inn lengden og høyden til en trekant og returnerer arealet til trekanten. Funksjonen skal kalle på funksjonen du lagde i oppgave a)
- d) Bruk funksjonen til å finne arealet av en trekant med høyde 14 og lengde 26

### Oppgave 3 *Fibonacci rekken*

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

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

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

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

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

$$\frac{a_n}{a_{n-1}} \rightarrow \phi. \quad (3)$$

- 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?

#### Oppgave 4 *Fakultet*

Si at du har 5 forskjellige små skulpturer du skal sette opp på rekke. Hvor mange ulike måter kan du gjøre dette på? For den første har du 5 valg, deretter har du 4 valg, så 3, og så videre. Dermed blir antall kombinasjoner til

$$5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120.$$

I matematikken skriver vi dette mer kompakt som  $5!$ , som vi kaller *fakultet*. Å rangere  $n$  ting kan altså gjøres på  $n!$  måter, som blir

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n.$$

For små tall er det enkelt å regne ut fakultet for hånd, men dersom  $n$  begynner å bli litt større vil dette ta fryktelig lang tid.

- a) Du skal nå skrive et program som kan regne ut fakultet for oss. Du kan enten prøve å gjøre dette helt selv, eller følge vår "steg-for-steg"-instrukser under.

- Spør brukeren om hva  $n$  er, lagre svaret i en variabel. Husk å konvertere svaret med `int(input())`.
  - Lag en variabel `fakultet`, som du gir verdien 1.
  - Lag en for-løkke som går fra 1 til  $n$ .
  - For hver iterasjon av løkka, gang `fakultet`-variabelen din med tallene du løkker over.
  - Skriv ut det endelige svaret så brukeren kan se det.
- b) Test at programmet ditt gir  $5! = 120$ .
- c) En vanlig kortstokk har 52 unike kort. Hvor mange mulige rekkefølger kan vi få om vi stokker en kortstokk?
- d) I noen kortspill legger man til 2 jokere i kortstokken, slik at det nå er 54 kort i kortstokken. Hvor mange måter blir det nå å stokke kortstokken på?

## 1.1 Naturfag

### Oppgave 5 *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  $\Omega$

### Oppgave 6 *Vekstfaktor*

I en petriskål lever en bakteriekultur som består av 1000 bakterier som formerer seg raskt nok til at mengden bakterier øker med 50% hver time. I denne

oppgaven 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?

## 1.2 Biologi

### Oppgave 7 *Hvor mye antibiotika i blodet*

Legen din har gitt deg 150 mg antibiotika, som du skal ta hver 24 time. Vi antar at all antibiotika går inn i blodet ditt, og at leveren fjerner 60% av antibiotikanivået i blodet hver 12 time. Likningen for hvor mye antibiotika du har i blodet etter  $n$  doser:

$$x_n = 0.40x_{n-1} + 150 \quad (4)$$

- a) Bruk en **for**-løkke til å skrive ut antibiotikainnholdet i blodet ditt for de første 14 døgne.
- b) Etter 14 døgn slutter du på antibiotikakuren. Skriv ut mengden antibiotika du har i blodet hvert døgn de neste to ukene.

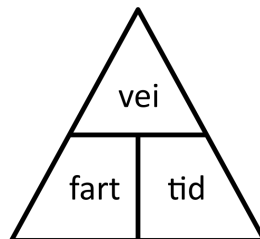
## 1.3 Fysikk

### Oppgave 8 *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 9 *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

$$[\text{kWh}] = 3.6[\text{MJ}], \quad (5)$$

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

$$[\text{cal}] = 4.2[\text{kJ}], \quad (6)$$

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?

## 1.4 Kjemi

### Oppgave 10 *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

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

- 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?

### Oppgave 11 *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?