Flere oppgaver

Her finner du flere relevante oppgaver – for alle de ulike temaene. Dersom du blir fort ferdig, kan du jobbe videre med disse, eller du kan bruke de for å øve deg mer på programmering i ettertid. Oppgavene her er ofte litt mer utfordrende enn de andre oppgavene vi har gitt i kurset, men vi håper at de også er mer interessante.

God koding!

1 Funksjoner

Oppgave 1 *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 2 *Kanonkule*

Året er 1384 i en fiktiv borg i Sør-England. Franskmennene har omringet borgen og kanonene er rullet frem. Dette har åpenbart gjort Lorden som eier borgen nervøs, og han spør deg, hans fremste lærdmann om kanonkulene vil nå borgen med den avstanden de har.

Heldigvis har spionene til kongen klart å stjele en av kanonene, og etter mye om og men, har du funnet ut at kulene forlater kanonen med en fart på ca 50 meter i sekundet. Kanonene er låst i en 30 graders vinkel, så posisjonen til en kanonkule kan beskrives med disse funksjonene

$$x(t) = 25\frac{m}{s}t$$
$$y(t) = 45\frac{m}{s}t - 5\frac{m}{s^2}t^2,$$

hvor $x(t)$ er hvor langt kanonkula har bevegd seg bortover (i meter) t sekund etter at den ble skutt og $y(t)$ er høyden til kanonkula (i meter) t sekund etter at den ble skutt.

a) Definer en funksjon `kanonkule_x(t)` som returnerer hvor langt kanonkula har bevegd seg bortover etter `t` sekund.

b) Definer en funksjon `kanonkule_y(t)` som returnerer høyden til kanonkula etter `t` sekund.

c) Definer en funksjon `kanonkule_posisjon(t)` som returnerer både hvor langt kula har bevegd seg bortover og høyden til den etter `t` sekund.

d) Kanonene er plassert 200 meter unna borgen (i x retning), og er i samme høyde som borgen. Bruk funksjonen vi akkurat lagde og prøv ut forskjellige verdier for `t`. Treffer kanonkula borgen?

e) (Bonusoppgave) Lag et program som bruker `input` funksjonen til å be om et antall sekunder og printer ut posisjonen til kanonkula så lang tid etter at den ble skutt.

f) (Bonusoppgave) Bruk en løkke til å printe ut posisjonen til kanonkula de ti første sekundene kanonkula er i lufta.

Oppgave 3 *Elekstrisitet: 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 (Ω). 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 Ω
- Resistansen når spenningen er 230 V og strømmen er 20 A
- Strømmen når spennignen er 5 V og motstanden er 200 Ω

c) Når flere motstander er koblet i seriekobling, blir den totale motstanden lik summen av alle motstandene i serien. $R_{tot} = R_1 + R_2 + \cdots + R_N$. 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 Ω er koblet i serie, hva blir strømmen da?

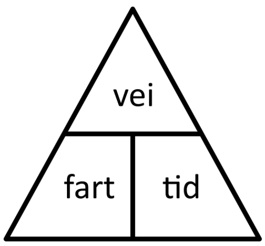
e) **Utfordring:** Når motstander er koblet i parallell, er sammenhengen mellom dem $\frac{1}{R_{tot}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_N}$. Lag en funksjon `parallellkobling` som regner ut den totale motstanden i kretsen. Hva blir strømmen nå, dersom motstandene og spenningen er den samme?

Oppgave 4 *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.

2 Plotting

2.1 Matematikk

Oppgave 5 *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:

```
from pylab import *

```

```python
x = arange(0, ..., 0.1)
A = ...

plot(x, A)
xlabel('x')
ylabel('Areal')
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 6 *Fortsettelse kanonkule*

Fortsettelse fra kanonkuleoppgaven. Vi har akkurat fortalt lorden i borgen vår at kanonene kan treffe og han tror ikke på oss. Derfor bestemmer vi oss for å lage en tegning hvor vi viser kulas bane. Om du ikke fikk til forrige kanonkuleoppgave kan du bruke denne funksjonen:

```python
def kanonkule_posisjon(t):
    x = 25*t                # kanonkule_x(t)
    y = 45*t - 5*t**2       # kanonkule_y(t)

    return x, y
```

a) Lag en array, `t` som inneholder en tallrekke som starter på 0, slutter på 9 og har en steglenge på 0.1 (Tips: hvis du vil at tallrekka skal slutte på 9 er det lurt å sette sluttposisjonen til `slutt+steglengde`, som i dette tilfellet er 9.1).

b) Bruk `kanonkule_posisjon(t)` til å regne ut kanonkulas `x`-koordinater og `y`-koordinater for hvert tidspunkt i `t` array-et vårt.

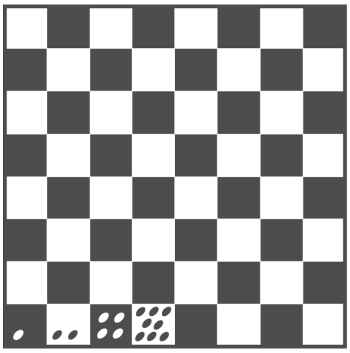
c) Lag et plot som viser `x`-posisjon på førsteaksen on `y`-posisjon på andreaksen.

d) Ved å løse likningen $x(t) = 200$ ser vi at kanonkula treffer borgen etter åtte sekund. Marker punktet $(x(8), y(8))$ med et rødt punkt.

Oppgave 7 *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!



a) Bruk en for-løkke til å finne ut hvor mange riskorn oppfinneren ba om. Er du enig med kassererens fortvilelse?

b) 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.

c) Lag et plot over veksten av riskorn. La x-aksen være antall sjakkruter 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()`.

Oppgave 8 *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 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

```
from pylab import arange

tallrekke = arange(10)*2
første_tall = tallrekke[0]
femte_tall = tallrekke[4]

print(første_tall)
print(femte_tall)
```

```
0
8
```

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

```
from pylab import arange

tallrekke = arange(4)*2
print(tallrekke)

tallrekke[0] = -1
tallrekke[2] = 0

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 plotte bakteriemengde på yaksen og timer på y-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?