Oppgaver

I denne seksjonen finner du oppgaver som hører til dag 1 av Kodeskolens kræsjkurs i programmering for lærere.

Tema for første dag er *introduksjon til Python*, *variabler*, *input*, *løkker* og *betingelser* Dersom du står fast er det bare å spørre. I tillegg anbefaler vi å lese i kompendiet hvis det er noen temaer du synes er spesielt vanskelige. Oppgaver markert som bonusoppgaver er litt mer utfordrende og du velger selv om du har lyst til å prøve deg på dem. Det er også lov å hoppe over en oppgave dersom den ikke er utfordrende nok. God koding!

Variabler og regning

Oppgave 1 *Printing*

a) Lag et program som skriver ut teksten «Hei, verden!», til skjermen.

b) Lag et program der du først lagrer navnet ditt i en variabel, og så få programmet ditt til å skrive ut en hilsen direkte til deg.

c) Hvis du bruker kommandoen `len()` på variabelen din får du ut antall bokstaver i navnet ditt. Endre programmet ditt så det også skriver ut denne informasjonen.

d) Du kan bruke funksjonen `input()` til å stille brukeren et spørsmål. Bruk dette til å lage et program som spør brukeren om navnet deres, og deretter skriver ut en beskjed som bruker navnet de har oppgitt. Skriv også ut hvor mange bokstaver det er i navnet til vedkommende.

Løsning oppgave 1 *Printing*

a)

```
print("Hei, verden!")
```

b)

```python
navn = "Maria"
print(f"Hei, {navn}!")
```

c)

```python
navn = "Maria"
print(f"'{navn}' har {len(navn)} bokstaver i seg."
    )
```

d)

```python
navn = input("Hva heter du?\n")
print(f"Hei, {navn}! Navnet ditt har {len(navn)}
    bokstaver i seg.")
```

Oppgave 2 *Kvadrattall*

Kvadrattall er heltall som er blitt kvadrert, altså ganget med seg selv, eller opphøyet i annen. I Python kan du regne ut kvadratet av et tall `n` enten ved å skrive `n*n` eller `n**2`.

a) Skriv et program som spør brukeren om et tall, og deretter skriver ut kvadratet av tallet brukeren ga.

Pass på at når vi spør om et tall må vi gjøre om svaret til fra brukeren til en tallvariabel ved å skrive `int` på følgende måte: `int(input())`. Dette er fordi «int» står for «integer», som er engelsk for heltall.

b) Bruk programmet ditt og prøve-feile metoden til å finne det minste tallet som har et kvadrat på over 1000.

Løsning oppgave 2 *Kvadrattall*

a)

```python
base = float(input("Skriv inn et tall:"))
```

```
kvadrat = base*base
print(f"Kvadratet av {base} er {kvadrat}")
```

b) Ved å bruke programmet og prøve-feile-metoden kan vi finne ut at det største tallet som har kvadrattall mindre enn 1000 er 31.

Oppgave 3 *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°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°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.

Løsning oppgave 3 *Konvertering av temperatur*

a)

```
fahrenheit = float(input('Fahrenheit: '))
celcius = (5/9)*(fahrenheit-32)

print(f'{fahrenheit} grader fahrenheit tilsvarer {celcius:.0f} celcius')
```

b)

```
Fahrenheit: 350
350.0 grader fahrenheit tilsvarer 177 celcius
```

177 ° C virker som en rimelig kakebaketemperatur

c)

$$F = \frac{9}{5}C + 32.$$

d)

```
celcius = float(input('Celcius: '))
fahrenheit = (9/5)*celcius + 32

print(f'{celcius} grader celcius tilsvarer {fahrenheit:.0f} fahrenheit')
```

e)

```
Celcius: 0
0.0 grader celcius tilsvarer 32 fahrenheit
```

```
Celcius: 100
100.0 grader celcius tilsvarer 212 fahrenheit
```

Oppgave 4 *Finn fire feil!*

Her følger det fire programmer som har blitt skrevet feil. Finn feilen i hver programsnutt. Du kan godt kjøre programmet inn på din egen maskin, og kjøre det, da kan kanskje feilmeldingen hjelpe deg å skjønne hva som er galt.

Når du tror du skjønner hva som er galt, rett feilen på din egen maskin, og kjør programmet for å sjekke at det fungerer som det skal.

a)

```python
print(Hei, Verden!)
```

b)

```python
Print("Hei, Verden!")
```

c)

```python
person = input("Hva heter du?")
print("Hei på deg, {navn}")
```

d)

```python
pi = 3,14
radius = 4
```

```python
areal = radius*pi**2
```

Løsning oppgave 4 *Finn fire feil!*

a) Vi må huske på fnuttene våre

```python
print("Hei, Verden!")
```

b) Vi må bruke liten p i print:

```python
print("Hei, Verden!")
```

c) Vi må passe på at vi bruker riktig variabel:

```python
navn = input("Hva heter du?")
print("Hei på deg, {navn}")
```

d) Vi må bruke punktum som desimaltegn, ikke komma:

```python
pi = 3.14
radius = 4
areal = radius*pi**2
```

Oppgave 5 *Radianer*

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

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

Hvor θ er vinkelen i radianer og ϕ er vinkelen i radianer

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 °

c) Modifiser programmet ditt så det spør brukeren om en vinkel i grader skriver ut hva det er i radianer.

d) Finn et utrykk 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.

Løsning oppgave 5 *Radianer*

a)

```
vinkel_grader = 45
```

b)

```
from math import pi

vinkel_grader = 45
vinkel_radianer = pi * vinkel_grader / 180
print(f'{vinkel_grader} grader tilsvarer {
   vinkel_radianer:.2f} radianer')
```

c)

```
from math import pi

vinkel_grader = float(input('Skriv inn en vinkel i
    grader: '))
vinkel_radianer = pi * vinkel_grader / 180
print(f'{vinkel_grader} grader tilsvarer {
   vinkel_radianer:.2f} radianer')
```

d)

```
from math import pi

vinkel_radianer = float(input('Skriv inn en vinkel
    i radianer: '))
vinkel_grader = vinkel_radianer*180/pi
```

```
print(f'{vinkel_radianer} radianer tilsvarer {
    vinkel_grader.2f} grader')
```

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 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?

Løsning oppgave 6 *Jordkloden*

a)

```
radiuskm = 6371
```

```python
volumkm = (4/3)*3.14*(radiuskm**3)
print("Volumet til jorden er", volumkm, "
   kubikkkilometer.")
```

b)

```python
radiusdm = 6371 * 10**4
volumdm = (4/3)*3.14*(radiusdm**3)
print("Volumet til jorden er", volumdm, "liter.")
```

c)

```python
massejord = 5.972 * 10**24
vektperliter = massejord/volumdm
print("I gjennomsnitt veier hver liter av jorda",
   vektperliter, "kg.")
```

Oppgave 7 *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 µm. Til slutt, lag en variabel som på ny definerer din høyde i cm, men regnet fra µm. Print denne siste variabelen, har du regnet rett og fått riktig høyde i cm?

Løsning oppgave 7 *Regne mellom SI-enheter*

```python
din_høyde_cm = 165
din_høyde_mm = din_høyde_cm * 10
din_høyde_um = din_høyde_mm * 10**3
din_høyde_nycm = din_høyde_um / 10**4

```

```
print(din_høyde_nycm)
```

Løkker

Oppgave 8 *Finne kvadrattall og kubikktall*

Lag et program som regner ut kvadrattallet og kubikktallet av tallene 1-5, og printer ut på en linje tallet og tilhørende kvadrat og kubikk. Bruk en `for`-løkke og legg til mellomrom mellom tallene.

Løsning oppgave 8 *Finne kvadrattall og kubikktall*

```
for tall in range(1, 6):
    kvadrat = tall**2
    kubikk = tall**3
    print(f'{tall:4} {kvadrat:4} {kubikk:4}')
```

Oppgave 9 *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?

Løsning oppgave 9 *Renter*

a)

```python
rente_bank_1 = 3/100
rente_bank_2_første_fem_år = 3.3/100
rente_bank_2_siste_fem_år = 2.8/100

inskudd = 10_000

penger_i_bank_1 = inskudd
penger_i_bank_2 = inskudd

for år in range(10):
    penger_i_bank_1 *= (1 + rente_bank_1)
    if år < 5:
        penger_i_bank_2 *= (1 + rente_bank_2_første_fem_år)
    else:
        penger_i_bank_2 *= (1 + rente_bank_2_siste_fem_år)

print(f"Etter 10 år står det {penger_i_bank_1:.2f} kr i bank 1, og {penger_i_bank_2:.2f} kr i bank 2.")
# >>> Etter 10 år står det 13439.16 kr i bank 1, og 13504.15 kr i bank 2.
```

b)

```python
rente_bank_1 = 3/100
rente_bank_2_første_fem_år = 3.3/100
rente_bank_2_siste_fem_år = 2.8/100

inskudd = 10_000
```

```python

penger_i_bank_1 = inskudd
penger_i_bank_2 = inskudd
år = 0
while penger_i_bank_1 <= penger_i_bank_2:
    penger_i_bank_1 *= (1 + rente_bank_1)
    if år < 5:
        penger_i_bank_2 *= (1 + rente_bank_2_første_fem_år)
    else:
        penger_i_bank_2 *= (1 + rente_bank_2_siste_fem_år)
    år += 1

print(f"Etter {år} år står det {penger_i_bank_1:.2f} kr i bank 1, og {penger_i_bank_2:.2f} kr i bank 2.")
#Etter 13 år står det 14685.34 kr i bank 1, og 14670.55 kr i bank 2.
```

c)

```python
rente_bank_1 = 3/100
rente_bank_2_første_fem_år = 3.3/100
rente_bank_2_siste_fem_år = 2.8/100

inskudd = 0
årlig_inskudd = 1000

penger_i_bank_1 = inskudd
penger_i_bank_2 = inskudd

for år in range(10):
    penger_i_bank_1 += årlig_inskudd
    penger_i_bank_2 += årlig_inskudd
    penger_i_bank_1 *= (1 + rente_bank_1)
    if år < 5:
        penger_i_bank_2 *= (1 + rente_bank_2_første_fem_år)
    else:
```

```
        penger_i_bank_2 *= (1 + rente_bank_2_siste_fem_år)

print(f"Etter 10 år står det {penger_i_bank_1:.2f} kr i bank 1, og {penger_i_bank_2:.2f} kr i bank 2.")
# >>> Etter 10 år står det 11807.80 kr i bank 1, og 11770.25 kr i bank 2.
```

d)

```
rente_bank_1 = 3/100
rente_bank_2_første_fem_år = 3.3/100
rente_bank_2_siste_fem_år = 2.8/100

inskudd = 1000
årlig_inskudd = 1000

penger_i_bank_1 = inskudd
penger_i_bank_2 = inskudd
år = 0
while penger_i_bank_1 <= penger_i_bank_2:

    penger_i_bank_1 *= (1 + rente_bank_1)
    if år < 5:
        penger_i_bank_2 *= (1 + rente_bank_2_første_fem_år)
    else:
        penger_i_bank_2 *= (1 + rente_bank_2_siste_fem_år)

    penger_i_bank_1 += årlig_inskudd
    penger_i_bank_2 += årlig_inskudd
    år += 1
print(f"Etter {år} år står det {penger_i_bank_1:.2f} kr i bank 1, og {penger_i_bank_2:.2f} kr i bank 2.")
# >>> Etter 9 år står det 11463.88 kr i bank 1, og 11449.66 kr i bank 2.
```

Oppgave 10 *Trekanttall*

Et trekanttall er summen av tallrekken

$$1, 2, \ldots, n.$$

For eksempel så er

$$1 + 2 + 3 + 4 + 5 = 15,$$

så da er 15 et trekanttall. Siden det var summen av de 5 første tallene i tallrekka, så sier vi at det er det *femte* trekanttallet.

La oss si at vi vil vite hva det hundrede trekanttallet er, da må vi legge sammen alle tallene fra 1 til 100. Det blir fort slitsomt og kjedelig å gjøre for hånd. Så la oss bruke programmering.

For å gjøre det lettere å sjekke om programmet vårt, så startet vi med å prøve å regne ut summen fra 1 til 5.

a) Lag en løkke som skriver ut tallene

$$1, 2, 3, 4 \text{ og } 5$$

til skjermen.

b) Endre nå løkka så du isteden finner summen av tallene 1 til 5. Da må du først lage en variabel utenfor løkka, og for hvert tall, legge det til variabelen din. Husk at du kan legge noe til en varibel med `+=`.

c) Sammenlign svaret programmet ditt gir med det vi fant for hånd. Er de to like? Hvis de ikke er det så er det noe galt!

d) Hvis programmet ditt fungerte som forventet kan du nå endre sånn at du regner summen av de første 100 tallene

$$1 + 2 + 3 + \ldots + 100.$$

Det er en kjent matematiker, Carl Friedrich Gauss, som fikk denne oppgaven av sin mattelærer når han gikk på skolen på 1700-tallet. Læreren tenkte nok at dette skulle holde Gauss opptatt en god stund med å legge sammen tall etter tall. Gauss hadde ikke tilgang til en datamaskin, så han kunne ikke automatisere jobben slik vi har gjort, men ha la merke til et mønster i tallene. Gauss la merke til at om vi starter på begge endene av rekka får vi et mønster

$$1 + 100 = 101, 2 + 99 = 101, 3 + 98 = 101, \ldots 50 + 51 = 101.$$

Fra dette mønsteret klarte Gauss å finne en formel for summen av tallene fra 1 til n, og uttrykket hans var

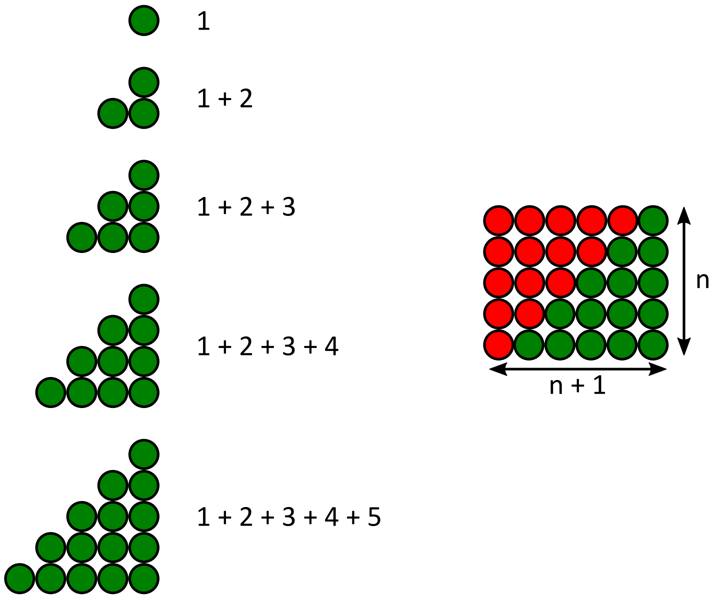
$$T_n = \frac{n(n+1)}{2}.$$

---

e) Bruk formelen til Gauss og sjekk at du får samme svar som programmet ditt for $n = 100$.

f) Gjør det samme for $n = 1000$, så $n = 1000000$ (én million).

For å skjønne hvorfor denne formelen er som den er, så kan det lønne seg å skjønne hvorfor de kalles trekanttall. Om vi tegner opp summene som antall baller, og tenger først 1, så 2, og så 3 bortover, sånn som dette:



Så ser vi at de ulike summene blir trekanter. Vi kan så gjøre om en slik trekant til en firkant ved å legge på like mange nye. Sånn som vist på høyre side av figuren. Denne firkanten har n baller i høyden, og $n + 1$ baller bortover. Da er antall baller i hele firkanten $n(n+1)$. Men vi har jo doblet antall baller for å få en firkant, så om vi bare skal telle de grønne ballene må vi dele på to.

---

g) Hvordan kan kan du sjekke om et tall er et trekanttall? Skriv et program som sjekker om et heltall er et trekanttall, og hvis ja, skriver ut faktorene i dette. For eksempel bør programmet skrive ut `1 + 2 + 3 + 4 + 5` hvis det blir gitt input `15`.

Løsning oppgave 10 *Trekanttall*

a) Dette gjør vi med en `for`-løkke, og `range`-funksjonen:

```
for tall in range(1, 6):
    print(tall)
```

Husk at `range` er fra-og-med, til (men ikke med), derfor skriver vi 1-6, for å få tallene 1, 2, 3, 4, og 5.

b)

```
total = 0

for tall in range(1, 6):
    total += tall

print(total)
```

Her må vi både opprette `total` før løkka, og printe den ut etter løkka. Vi ser hvilke kodelinjer som hører til løkka fordi de har fått innrykk. I tilegg har vi lagt til blanke linjer for å skille dem litta fra løkka, men merk at dette er frivillig.

c) Svaret blir 15, som forventet.

d) Vi endrer programmet ved å endre hva løkka går til. For å gå opp til og med hundre må vi skrive `range(1, 101)`. For å gjøre programmet vårt lettere å endre velger vi derimot å lage n som en variabel:

```
n = 100

total = 0
for tall in range(1, n+1):
    total += tall

print(total)
```

Det er nå rett-frem å endre programmet, bare ved å endre den første linja.

Svaret blir 5050

e) Vi fikk 5050 for $n = 100$, formelen gir

$$\frac{n(n+1)}{2} = \frac{100 \cdot 101}{2} = 5050.$$

Som altså er det samme, programmet vårt ser ut til å fungere.

f) For $n = 1000$ gir programmet vårt oss 500500. Formelen gir oss det samme. For $n = 1000000$ gir programmet vårt oss 500000500000, og formelen gir igjen det samme.

Betingelser

Oppgave 11 *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 $\mid a \mid$ og er definert som:

$$\mid a \mid = \begin{cases} a, & \text{hvis } a \geq 0 \\ -a, & \text{hvis } < 0 \end{cases} \tag{1}$$

a) Lag et progam som ber brukeren om et tall og skriver ut absoluttverdien av tallet

b) Lag et program som ber brukeren om to tall og skriver ut hvilket av tallene som har høyest absoluttverdi

Løsning oppgave 11 *Absoluttverdi*

a)

```python
tall = float(input('Skriv inn et tall'))

if tall < 0:
    absoluttverdi = -tall
else:
    absoluttverdi = tall

print(absoluttverdi)
```

b)

```python
tall1 = float(input('Skriv inn et tall'))
tall2 = float(input('Skriv inn enda et tall'))

if tall1 < 0:
    absoluttverdi1 = -tall1
else:
    absoluttverdi1 = tall1

if tall2 < 0:
    absoluttverdi2 = -tall2
else:
    absoluttverdi2 = tall2

if absoluttverdi1 > absoluttverdi2:
    print(absoluttverdi1)
else:
    print(absoluttverdi2)
```

Oppgave 12 *ABC-formelen*

ABC-formelen for å løse annengradsformler er som følger:

La a, b og c være reelle tall, hvor $a \neq b$. Da har likningen $ax^2 + bx + c = 0$ løsningene

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

a) Lag et progam som finner løsningene til en annengradslikning med $a = 1$, $b = 2.5$ og $c = 1$.

b) Modifiser programmet ditt så det spør brukeren om verdier for a, b og c og skriver ut de tilhørende løsningene.

c) Dersom $b^2 - 4ac < 0$ har den tilhørende annengradslikningen ingen reell løsning. Bruk en `if`-betingelse til å informere brukeren om at det ikke finnes noen reell løsning dersom $b^2 - 4ac < 0$

d) Dersom $b^2 - 4ac = 0$ har likningen kun en løsning. Bruk en `elif` til å sjekke om dette er tilfelle og i så tilfelle informere brukeren om at det kun er en løsning

Løsning oppgave 12 *ABC-formelen*

a)

```
from math import sqrt

a = 1
b = 2.5
c = 1
løsning1 = (-b + sqrt(b**2 - 4*a*c))/(2*a)
løsning2 = (-b - sqrt(b**2 - 4*a*c))/(2*a)

print(f'Løsningene for likningen {a}x^2 + {b}x + {
    c} = er {løsning1} og {løsning2}')
```

b)

```
from math import sqrt

a = float(input('Hva er a? '))
b = float(input('Hva er b? '))
c = float(input('Hva er c? '))

løsning1 = (-b + sqrt(b**2 - 4*a*c))/(2*a)
løsning2 = (-b - sqrt(b**2 - 4*a*c))/(2*a)

print(f'Løsningene for likningen {a}x^2 + {b}x + {
    c} = er {løsning1} og {løsning2}')
```

c)

```
from math import sqrt

a = float(input('Hva er a? '))
b = float(input('Hva er b? '))
```

```
c = float(input('Hva er c? '))

rot_del = b**2 - 4*a*c

if rot_del < 0:
    print(f' {a}x^2 + {b}x + {c} = 0 har dessverre ingen reelle løsninger')
else:
    løsning1 = (-b + sqrt(rot_del))/(2*a)
    løsning2 = (-b - sqrt(rot_del))/(2*a)
    print(f'Løsningene for likningen {a}x^2 + {b}x + {c} = 0 er {løsning1} og {løsning2}')
```

d)

```
from math import sqrt

a = float(input('Hva er a? '))
b = float(input('Hva er b? '))
c = float(input('Hva er c? '))

rot_del = b**2 - 4*a*c

if rot_del < 0:
    print(f' {a}x^2 + {b}x + {c} = 0 har dessverre ingen reelle løsninger')
elif rot_del == 0:
    løsning = -b/(2*a)
    print(f'Løsningen for likningen {a}x^2 + {b}x + {c} = 0 er {løsning}')
else:
    løsning1 = (-b + sqrt(rot_del))/(2*a)
    løsning2 = (-b - sqrt(rot_del))/(2*a)
    print(f'Løsningene for likningen {a}x^2 + {b}x + {c} = 0 er {løsning1} og {løsning2}')
```