

# Oppgaver og Løsningsforslag

## Eksamensutdrag 2023

Eksamensoppgaver (og eksempeloppgaver) med Python fra 1P og 2P 1T

### Oppgave3 -1P H21 del 1

Forklar hva som skjer når programmet nedenfor kjøres.

Hva forteller de to tallene som skrives ut i linje 10 og 11(print verdi og print år)

```
beløp = 10000
verdi = beløp
vekstfaktor = 1.03
år = 0

while verdi < beløp * 2:
    verdi = verdi * vekstfaktor
    år = år + 1

print(verdi)
print(år)
```

I starten legger man opp variabler man kan kalle på i programmet.  
I while løkken legger man inn faktoren så lenge verdi er mindre en beløpet ganget med 2, tell antall år til grenseverdien oppnås (år = år+1). Den vil gå litt over siden print funksjonen kommer etter utførelsen av verdi ganget med vekstfaktor. Vekstfaktor er rentesats.

Legger man derimot print funksjonen av verdi under while, vil den stoppe før den når 2 gangen av verdi.

Svar slik den er satt opp i eksemplet.  
20327.941064604045  
24

### Oppgave 5 -1P V22 del 1

```
startverdi = 2000
verdi = startverdi
vekstfaktor = 1.05
år = 0

while verdi < startverdi * 2:
    verdi = verdi * vekstfaktor
    år = år + 1

print(verdi)
print(år)
```

I starten legger man opp variabler man kan kalle på i programmet.  
I while løkken legger man inn faktoren så lenge verdi er mindre en beløpet ganget med 2, tell antall år til grenseverdien oppnås (år = år+1). Den vil gå litt over siden print funksjonen kommer etter utførelsen av verdi ganget med vekstfaktor. Vekstfaktor er her tenkt som rentesats.

Legger man derimot print funksjonen av verdi under while, vil den stoppe før den når 2 gangen av verdi.

Eleven ønsker å finne ut hvor mange år det tar før innskuddet hans på 2000 har doblet seg når rentesatsen er 5%

Svar slik den er satt opp i eksemplet.  
4157.856358822737  
15

# Oppgaver og Løsningsforslag

## Eksamensutdrag 2023

### Oppgave 5 -1P H22 del 2

Lars ønsker å finne funksjonen av  $x$ . Når han kjører programmet får han 5 som løsning på funksjonen.

```
def f(x):  
    return 3 * x - 15  
x = 0  
while x <= 10:  
    if f(x) == 0:  
        print(x)  
        x = x + 1
```

```
def f(x):  
    return x**2 - 6*x + 8  
x = 0  
while x <= 10:  
    if f(x) == 0:  
        print(x)  
        x = x + 1
```

```
def f(x):  
    return x**2 - 144  
x = 0  
while x <= 20:  
    if f(x) == 0:  
        print(x)  
        x = x + 1
```

Funksjonen  $f$  avhenger av variabelen  $x$ . `Return` avslutter funksjonen og returnerer verdien til det som står etter nøkkelordet `return`.

$x$  er satt til 0, som en teller variabel som teller seg opp til man finner en verdi gitt ut fra while løkken. I dette tilfellet sier man at så lenge  $x$  er mindre eller lik 10 skal løkka gå `while x <= 10:`. Hvis funksjonen av  $x$  blir lik 0 `if f(x) == 0:`, skriv ut  $x$  `print(x)`

Så legger vi inn en teller som teller opp  $x$  variabelen med en, hver gang `x = x + 1` frem til verdien av  $x$  gjør at uttrykket blir 0. Resultatet av programmet blir 5,  $5*3$  er 15 og  $15-15$  er 0.

Når Lars endrer funksjonsuttrykket til  $x^2 - 6x + 8$  blir svaret 2 og 4.

$$2x^2 - 6x + 8 = 0 \rightarrow 4 - 12 + 8 = 0$$

$$4x^2 - 6x + 8 = 0 \rightarrow 16 - 24 + 8 = 0$$

Når Lars endrer funksjonsuttrykket til  $x^2 - 144$  må han også endre verdien til  $x$  i while løkken, slik at den står i forhold til tallene vi arbeider med. I dette tilfellet er en av verdien 144 og 10 som står der fra tidligere oppgaver er maks. 100 ( $10^2$ ). Så eksempelvis kan man sette 15 eller 20 som er h.h.v. 225 og 400- dvs. godt innfor.

Kjører programmet en gang til og får 12.  $12^2$  gir 144 og løser  $x$  i forhold til 0.

# Oppgaver og Løsningsforslag

## Eksamensutdrag 2023

### Oppgave 2 -2P Eks H21 del 1

```
pris = 200000
verdi = pris
vekstfaktor = 0.85
år = 0
```

```
while verdi > pris / 2:
    verdi = verdi * vekstfaktor
    år = år + 1
```

```
print(verdi)
print(år)
```

I starten legger man opp variabler man kan kalle på i programmet.

I while løkken legger man inn faktoren så lenge verdi er halve beløpet / 2, telles antall år til grenseverdien oppnås (år = år+1). Den vil gå litt under halve siden print funksjonen kommer etter utførelsen av verdi ganget med vekstfaktor. Vekstfaktor er her tenkt som rentesats.

Legger man derimot print funksjonen av verdi under while, vil den stoppe før den når den halve verdien.

Eleven ønsker å finne ut hvor mange år det tar før vekstfaktoren halverer prisen til en enhet når vekstfaktoren er 0,85. Eksempelvis nedskrivning av en bil.

Svar slik den er satt opp i eksemplet.  
5 år

### Oppgave 4 -2P v22 del 1

```
beløp = 0
vekstfaktor = 1.02
innskott = 20000
år = 0
```

```
while beløp < 500000:
    beløp = beløp + innskott
    beløp = beløp * vekstfaktor
    år = år + 1
```

```
print(år)
print(beløp)
```

I starten legger man opp variabler man kan kalle på i programmet.

I while løkken legger man inn faktoren så lenge beløp er mindre en <500000, telles antall år til grenseverdien oppnås (år = år+1). Men den økes med innskudd på 20.000 hvert år i tillegg til vekstfaktoren som eksempelvis kan være en rentesats.

Legger man derimot en print funksjon under hver av verdiene **beløp = beløp + innskott** og **beløp = beløp \* vekstfaktor** vil man se hvilken grad hver av faktorene påvirker sparingen frem til ferdig spåremål.

Ada ønsker å finne ut hvor mange år det tar før hun har oppnådd egenkapitalen på 500.000, ved hjelp av 20.000 i innskudd pr. år med en rentesats på 2%.

Svar slik den er satt opp i eksemplet.  
21  
525979.6707758801

# Oppgaver og Løsningsforslag

## Eksamensutdrag 2023

### Oppgave -2P H22 del 1

```
a = 4  
b = 5  
c = 3
```

```
if a**2 + b**2 == c**2 or a**2 + c**2 == b**2 or b**2 + c**2 == a**2:
```

```
    print('Kontrollutregningen av en rettvinklet trekant er riktig!')  
else:  
    print(' Dette er ikke en rettvinklet trekant.', '\n', 'Se over målene, se at de blir riktige?')
```

Lars har laget et program for rettvinklet trekant ved hjelp av Pythagoras læresetning. Når dataene er riktige kan programmet, eksempelvis, skrive ut at kontrollutregningen stemmer eller at den er feil. Programmet kan også utvides og lagre dataene for å vise dem i en Utskrift senere.

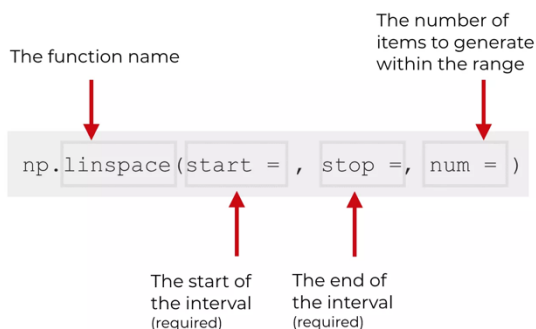
### Oppgave -1T eks 2020 del 1

```
import numpy as np  
  
def f(x):  
    return x**2 - 2  
  
x = np.linspace (-2,2,400)  
  
y = f(x)  
  
for i in range (0,399):  
    print(i)  
    if y[i] * y[i+1]<= 0:  
        print('Jeg har funnet nullpunktet')
```

I starten legger henter man opp et bibliotek og definerer en del av dette, `linspace` -funksjonen oppretter sekvenser med jevnt fordelte verdier innenfor et satt intervall. Du angir et startpunkt og et slutt punkt for intervallet, og deretter angir du totalt antall avbruddspunkter du vil ha innenfor dette intervallet (inkludert start- og slutt punkt).

`x = np.linspace (-2,2,400)`. Første tall `-2` angir start av intervall, minste verdi, `2` er siste og største tall i intervallet og `400` er antall punkter, tall som vi skal ha i tabellen.

Svar slik den er satt opp i eksemplet. Svaret, jeg har funnet nullpunktet, kommer opp 2 ganger, dvs. 2 svar i funksjonen  $x^2-2$



# Oppgaver og Løsningsforslag

## Eksamensutdrag 2023

```
import numpy as np

def f(x):
    return x**2 - 2

x = np.linspace (-2,2,400)

y = f(x)

for i in range (0,399):
    #print(i)
    if y[i] * y[i+1]<= 0:
        print('Jeg har funnet nullpunktet')
        print(i)
```

Legger inn først en hjelpeutskrift for i in range, slik at jeg kan se at tall og 0 punkt bekreftelse kommer sammen, og så en utskrift av de respektive verdiene i listen[ i ] og [i+1]dvs. første og siste verdi for nullpunkt.

Svar slik den er satt opp i eksemplet.

```
Jeg har funnet nullpunktet
Nullpunktet: 58
Jeg har funnet nullpunktet
Nullpunktet: 340
```

Monica kjører programmet og får ut 2 linjer med riktige svar, dvs. 2 løsninger for nullpunkt. Programmet er utvidet for å kunne skrive ut verdien når den tilnærmede verdien for nullpunktet oppstår.

### Oppgave 5 -1T eks H21 del 1

```
a = 1
b = 2
c = 1

d = (b**2) - (4 * a * c)

if d < 0:
    print('Vi har ingen løsning på ligningen.')
elif d == 0:
    x = -b / (2 * a)
    print("Den eneste løsningen for x = ", x)
else:
    x1 = (-b + math.sqrt(d)) / (2 * a)
    x2 = (-b - math.sqrt(d)) / (2 * a)
    print("Løsningen på x = ", x1, " and x = ", x2)
```

Programmet har 3 variabler med satt verdi, hhv. a1, b2 og c1.

Programmet vil først sjekke om verdien av d er mindre en 0. Hvis den er mindre en 0 vil programmet skrive ut «Vi har ingen løsning på ligningen.».

Hvis verdien av d er lik 0,  $d = (b**2) - (4 * a * c)$ - blir 0, da  $b=2$  vil programmet beregne og skrive ut den eneste løsningen for x.

Hvis d er større enn 0, vil programmet beregne og skrive ut de to mulige løsningene, x1 og x2, som er løsningene av en kvadratisk ligning.

Malin har nå lagt inn følgende i linje 8, 10 og 12.

Linje 8:  
print('Vi har ingen løsning på ligningen.')

Linje 10:  
x = -b / (2 \* a)  
print("Den eneste løsningen for x = ", x)

Linje 12:  
x1 = (-b + math.sqrt(d)) / (2 \* a)  
x2 = (-b - math.sqrt(d)) / (2 \* a)  
print("Løsningen på x = ", x1, " and x = ", x2)

Svar slik den er satt opp i eksemplet.

```
Den eneste løsningen for x er : -1
```

# Oppgaver og Løsningsforslag

## Eksamensutdrag 2023

### Oppgave 3 -1T eks H 21 del 2

```
def f(x):  
    return x**2-2  
  
a = -2  
e = 0.01  
  
while a < 2:  
    if f(a) * f(a+e) <= 0:  
        print('jeg har funnet et nullpunkt.')  
        #print(a)  
    a = a + e  
    #print(a)
```

I første del har vi satt variablene a og e. e er graderingen av trinnene og skal telles opp fra -2 til 2 - trinnvis med 0.01 i stegene. Så lenge a er mindre en 2 skal loopen i programmet gå. Når Nullpunkt finnes, skal dette skrives ut og teksten «Jeg har funnet et Nullpunkt» skal skrives til skjerm.

NB! Har lagt inn 2 hjelpeutskrifter for å analysere når nullpunkt inntreffer, se rød farge. Den første viser kun de aktuelle punktene. Den andre hele matrisen.

Når man aktiverer **#print(a)**, fjerner man bare tegnet «#» og lar print stå slik, **print(a)**. Da vil du få skrevet ut de 2 tilnærmede verdiene med 16 desimaler.

Leg inn følgende endring i print funksjonen: **print(f'{a:.4f}')**. Da skriver du ut med 4 desimalers nøyaktighet.

```
Svar slik den er satt opp i eksemplet.  
jeg har funnet et nullpunkt.  
Nullpunkt: -1.4199999999999995  
jeg har funnet et nullpunkt.  
Nullpunkt: 1.4100000000000026  
Svar med 4 desimalers nøyaktighet.  
jeg har funnet et nullpunkt.  
-1.4200  
jeg har funnet et nullpunkt.  
1.4100
```

### Oppgave 4 -1T V22 del 1

```
def f(x):  
    return x**2  
  
x = 1  
  
while f(x) <= 400:  
    print(f(x))  
    #print(x)  
    x = x + 1
```

Hjelpelinjen , **#print(x)**, gir indikasjon av Hvilke tall i teller og resultatlinjen.

Koden defineres en funksjon f(x). Programmet får en input av x og returnerer kvadratet.

Variabelen x settes til 1 og så går programmet inn i en while loop. Denne er satt til å løpe så lenge funksjonen av x er mindre en, eller lik, 400.

Så skriver programmet resultatene til skjermen.

Telle-funksjonen  $x = x+1$  teller opp, starter på 1, 2, 3 osv. til verdien er lik eller mindre en 400.

```
Svar slik den er satt opp i eksemplet.  
1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121  
,144, 169, 196, 225, 256, 289, 324, 361  
,400
```

Starter på  $1^2$  til og med  $20^2$

# Oppgaver og Løsningsforslag

## Eksamensutdrag 2023

### Oppgave 3 -1T H22 del 1

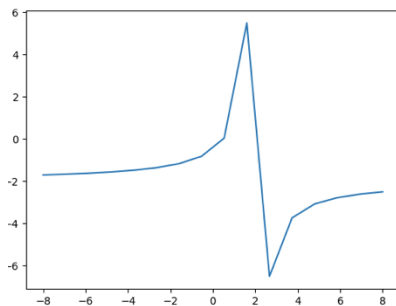
```
def f(x):  
    return (1-2 * x) / (x - 2)  
x = 8  
while x >= -8:  
    print(x,f(x))  
    x = x - 1
```

### Korrigert program

```
def f(x):  
    if x == 2:  
        return 'Kan ikke dele på null'  
    else:  
        return (1-2 * x) / (x - 2)  
  
x = 8  
  
while x >= -8:  
    print(x,f(x))  
    x = x - 1
```

### Skrive ut plott

```
from pylab import *  
  
def f(x):  
    return (1-2 * x) / (x - 2)  
  
x = linspace(-8,8,16)  
y = f(x)  
  
x = x - 1  
  
plot(x,f(x))  
show()
```



Koden defineres en funksjon  $f(x)$ . Programmet får en input av  $x$  og returnerer verdien av uttrykket  $(1-2 * x) / (x - 2)$ .

Variabelen  $x$  settes til 8 og så går programmet inn i en while loop. Denne er satt til å løpe så lenge funksjonen av  $x$  er mindre en, eller lik,  $-8$ .

Så skriver programmet resultatene til skjermen.

Telle-funksjonen  $x = x - 1$  teller ned, starter på 8, 7, 6 osv. til verdien er lik eller mindre en  $-8$ . Problemet når du kommer til  $x = 2$  vil man forsøke å dele med 0, og programmet faller ut med feilmelding.

For å korrigere på dette lager vi en ny løkke med if- else. I løkken sier vi at hvis tallet  $x=2$  kommer, skriv ut «kan ikke dele på null». Ellers skal du fortsette løkken ned til grenseverdien som er satt til  $-8$ , dvs. fra  $x = 8$  til  $x = -8$ .

```
8 -2.5  
7 -2.6  
6 -2.75  
5 -3.0  
4 -3.5  
3 -5.0  
2 Kan ikke dele på null  
1 1.0  
0 -0.5  
-1 -1.0  
-2 -1.25  
-3 -1.4  
-4 -1.5  
-5 -1.5714285714285714  
-6 -1.625  
-7 -1.6666666666666667  
-8 -1.7
```

