

Løsningsforslag for «Pythonprogrammering for Ungdomsskole og VGS»

Jeg har også lagd videoløsningsforslag til disse oppgavene. De kan du finne ved å følge linken eller QR-koden til oppgaven. Dersom du sliter med å forstå konsepter i oppgavene så er det anbefalt å benytte seg av videoløsning fortrinnsvis.

NB: Selv om programmet ditt er litt annerledes enn det som står her så betyr det ikke at ditt program er feil. Sjekk selv om du har oppfylt oppgaven ved å lese den godt.

Dette er en tidlig utgave av løsningsforslaget og alle løsningene er ikke perfekte. Om du ønsker en grundigere gjennomgang se videoløsningsforslag.

Oppgave 1 (Kommentar.py)

Program:

```
8
9      # a)
10
11      # Python leser ikke kommentarer
12
13      # b)
14
15      """
16      Velkommen til programmering!
17      Gjør ditt beste!
18      """
```

Utskrift (Output):

Dette programmet får ingen utskrift når det kjøres.

Oppgave 2 (print.py)

Program:

```
# a)

print("Jeg har laget mitt første program")

# b)

print(71)

# c)

print("3 + 2 =",5)
```

Utskrift (Output):

```
Jeg har laget mitt første program
71
3 + 2 = 5
```

Oppgave 3 (feilmeldinger.py)

Program:

a)

```
# a)

# Første bit av koden er det ikke noe galt med.
print("Hei")
# I koden under har jeg lagt inn en feil
print(Verden)

b)


```

Utskrift (Output):

a)

```
Hei
Traceback (most recent call last):
  File ~/Applications/Spyder.app/Contents/Resources/lib/python3.9/
py3compat.py:356 in compat_exec
    exec(code, globals, locals)
  File ~/Python oppgaver løsning/Python 1-10/feilmeldinger.py:14
    print(Verden)
NameError: name 'Verden' is not defined
```

b)



```
# b)

# Første bit av koden er det ikke noe galt med.
print("Hei")
# I koden under har jeg lagt inn en feil
print("Verden")
```

c) Alternativ 1) (Syntax error)

```
# c)

Første bit av koden er det ikke noe galt med.
print("Hei")
# I koden under har jeg lagt inn en feil
print("Verden")
```

Alternativ 2) (Intendent error)

```
Første bit av koden er det ikke noe galt med.
print("Hei")
# I koden under har jeg lagt inn en feil
print("Verden")
```

```
Hei
Verden
```

c) Alternativ 1 (Syntax error)

```
File <unknown>:11
  Første bit av koden er det ikke noe galt med.
    ^
SyntaxError: invalid syntax
```

Alternativ 2 (intendent error)

```
Første bit av koden er det ikke noe galt med.
    ^
IndentationError: unexpected indent
```

Oppgave 4 (variabler.py)

Program:

```
# a)

a = 4
b = 47.35
c = "Dette er gøy"

# b) -----

d = [a,b,c]

# c) -----

print(a)
print(b)
print(c)
print(d)
```

Utskrift (Output):

```
4
47.35
Dette er gøy
[4, 47.35, 'Dette er gøy']
```

Oppgave 5 (regneoperatorer.py)

Program:

```
# a) -----

a = 4
b = 5

# b) -----

c = a + b

# c) -----

print(c)

# d) -----

d = a - b
e = a * b
f = b / a

print(d)
print(e)
print(f)
```

Utskrift (Output):

```
9
-1
20
1.25
```



Oppgave 6 (streng_pluss_gange.py)

Program:

```
# a) -----
a = "Å legge sammen"
b = " "
c = "tekststrenger er litt rart"

# b) -----
print(a+c)
print(a+b+c)

# c) -----
print(5*c)

# d) -----
print("Nå "+"programmerer"+"jeg")

# d) Forbedring
print("Nå "+" programmerer "+"jeg")
```

Utskrift (Output):

```
Å legge sammentekststrenger er litt rart
Å legge sammen tekststrenger er litt rart
tekststrenger er litt rarttekststrenger er litt rarttekststrenger er litt
rarttekststrenger er litt rarttekststrenger er litt rart
Nåprogrammererjeg
Nå programmerer jeg
```

Oppgave 7 (eksponenter_heltall_mod.py)

Program:

```
# a) -----
print(5**3) # 5*5*5 =125
print(2**8) # 2^8 = 2*2*2*2*2*2*2*2=256

# b) -----
print(9//4) # 4 går 2 hele ganger i 9

# c) -----
print(9 % 4) # Når gjør divisjonen over får vi 1 i rest.
```

Utskrift (Output):

```
125
256
2
1
```

Oppgave 8 (konsolloppgave)

Program:

```
In [1]: print("Hallo verden")
Hallo verden

In [2]: 5 + 3
Out[2]: 8

In [3]: a = 5

In [4]: a+5
Out[4]: 10

In [5]: 4**2
Out[5]: 16

In [6]: 20//3
Out[6]: 6

In [7]: 20%3
Out[7]: 2
```

Utskrift (Output):



Oppgave 9 (dataendringer.py)

Program:

a)

```
# a) -----  
  
a = "42"  
b = 3.1  
print(a+b)
```

b)

```
# b) -----  
  
a = "42"  
b = 3.1  
  
a = int(a)  
print(a+b)
```

Oppgave 10 (input_kopi.py)

Program:

```
a = 5  
b = input("Skriv inn et tall: ")  
b = int(b)  
print(b)  
print(a+b)
```

Oppgave 11 (input_selv.py)

Program:

a) b) og c)

```
# a) -----  
  
a = input("Skriv inn ett heltall: ")  
  
b = 3.2 # input("Skriv inn ett flyttall: ")
```

```
# b) -----  
a = int(a)
```

```
# c) -----  
# b = float(b)  
print(a+b)
```

d)

```
# a) -----  
  
a = input("Skriv inn ett heltall: ")  
b = input("Skriv inn ett flyttall: ")  
  
# b) -----  
a = int(a)  
  
# c) -----  
# b = float(b)  
print(a+b)
```

e)

Utskrift (Output):

a)

```
Traceback (most recent call last):  
  File ~/Applications/Spyder.app/Contents/Resources/lib/python3.9/spy3compat.py:356 in compat_exec  
    exec(code, globals, locals)  
  File ~/Python oppgaver løsning/Python 1-10/dataendringer.py:13  
    print(a+b)  
TypeError: can only concatenate str (not "float") to str
```

b)

45.1

Utskrift (Output):

Skriv inn et tall:

Vi skriver inn 23 og trykker Enter

Skriv inn et tall: 23

23

28

Utskrift (Output):

a) b) og c)

Skriv inn ett heltall: 10

13.2

d)

```
Skriv inn ett heltall: 10  
Skriv inn ett flyttall: 2.3  
Traceback (most recent call last):  
  File ~/Applications/Spyder.app/Contents/Resources/lib/python3.9/spy3compat.py:356 in compat_exec  
    exec(code, globals, locals)  
  File ~/Python oppgaver løsning/Python 11-20/input_selv.py:19  
    print(a+b)  
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Her får vi en typefeil ettersom vi prøver å legge sammen ett heltall (integer) 10 og en tekststreng «2.3»

e)

Skriv inn ett heltall: 10

Skriv inn ett flyttall: 2.3

12.3



```
# a) -----
a = input("Skriv inn ett heltall: ")
b = input("Skriv inn ett flyttall: ")

# b) -----
a = int(a)

# c) -----
b = float(b)
print(a+b)
```

Oppgave 12 (input_fruktkiosk.py)

Program:

```
# a) -----
pris_kilo_eple = 60
pris_kilo_banan = 45

# b) -----
kg_eple = input("Skriv inn antall kg epler Knut kjøper: ") # Tar imot input
kg_eple = float(kg_eple) # Gjør om ifra strengverdi til flyttallsverdi
# Vi kan også gjøre begge deler i en kommando som under.
kg_banan = float(input("Skriv inn antall kg bananer Knut kjøper: "))

# c) -----
kostnad_epler = kg_eple * pris_kilo_eple
kostnad_bananer = kg_banan * pris_kilo_banan

# d) -----
print("Eplene koster til sammen:", kostnad_epler, "kroner")
print("Bananene koster til sammen:", kostnad_bananer, "kroner")
print("Den samlede prisen blir:", kostnad_epler+kostnad_bananer, "kroner")
```

Utskrift (Output):

```
Skriv inn antall kg epler Knut kjøper: 5
Skriv inn antall kg bananer Knut kjøper: 10
Eplene koster til sammen: 300.0 kroner
Bananene koster til sammen: 450.0 kroner
Den samlede prisen blir: 750.0 kroner
```

Oppgave 13 (konsolloppgave)

Program:

```
In [1]: 3==4
Out[1]: False

In [2]: 4==4
Out[2]: True

In [3]: "Hei" != 3
Out[3]: True

In [4]: 5 != 9
Out[4]: True

In [5]: 4>4
Out[5]: False

In [6]: 4<=8
Out[6]: True

In [7]: 99 != 98+1
Out[7]: False

In [8]: 100+33 >= 98+35
Out[8]: True
```

Utskrift (Output):



Oppgave 14 (sammenlikning.py)

Program:

```
# a) -----
print("Svaret på a er:")
print(3 + 3 == 6)

# b) -----
print("Svaret på b er:", 9**2 == 88 )

# c) -----
print("Svaret på c er:", 5 < 3 * 3 - 2 )

# d) -----
print("Svaret på d er:", 4**2 != 3 * 5 + 1 )

# e) -----
print("Svaret på e er:", -11 <= (-5)**2)

# f) -----
print("Svaret på f er:", 6**3 >= 333 )

# Strengverdier

# g) -----
print("Svaret på g er:", "Ja" == "Ja" )

# h) -----
print("Svaret på h er:", "Nei" == "Ja" )

# i) -----
print("Svaret på i er:", "Ja" == "ja" )

# j) -----
# I Python er det forskjell på stor og liten bokstav.
```

Utskrift (Output):

```
Svaret på a er:
True
Svaret på b er: False
Svaret på c er: True
Svaret på d er: False
Svaret på e er: True
Svaret på f er: False
Svaret på g er: True
Svaret på h er: False
Svaret på i er: False
```

Oppgave 15 (if_else_kopi.py)

Program:

```
a)
x = 2

if x == 3:
    print("Yes")
else:
    print("No")

b)
x = 3

if x == 3:
    print("Yes")
else:
    print("No")
```

Utskrift (Output):

```
a) _
No
b) _
Yes
```

Oppgave 16 (if_else_vq.py)

Program:

```
# a) -----
v = 400
q = 900

# b) -----
if q > v:
    print("q er størst")

# c) -----
else:
    print("q er mindre enn eller lik v")
```

Utskrift (Output):

```
b)
q er størst
d)
q er mindre enn eller lik v
```



d) (Bytter om på verdiene til v og q)

```
# a) -----  
v = 900  
q = 400  
  
# b) -----  
if q > v:  
    print("q er størst")  
# c) -----  
else:  
    print("q er mindre enn eller lik v")
```

Oppgave 17 (if_fin_dag.py)

Program:

```
# a) -----  
x = "Ja"  
if x == "Ja":  
    print("Det var godt å høre.")  
# b) -----  
else:  
    print("Det var trist.")  
  
c)  
# c) -----  
x = input("Har du en fin dag? Svar: ")  
if x == "Ja":  
    print("Det var godt å høre.")  
# b) -----  
else:  
    print("Det var trist.")
```

Utskrift (Output):

a) og b) Det var godt å høre.
c)
Har du en fin dag? Svar: Nei...
Det var trist.

Oppgave 18 (penger.py)

Program:

```
# a) -----  
penger = 200  
pris = 150  
# b) -----  
if penger > pris:  
    print("Einar har råd til fotballen og har", penger - pris, "kroner igjen.")  
# c) -----  
else:  
    print("Einar mangler", pris - penger, "kroner for å ha råd til fotballen")  
# d --- Kjør programmet for forskjellige verdier.
```

d) (Setter inn andre verdier for penger og pris)

```
penger = 200  
pris = 350
```

Utskrift (Output):

a)
Einar har råd til fotballen og har 50 kroner igjen.

d)
Einar mangler 150 kroner for å ha råd til fotballen

e), f) og g)
Lille Einar skal på lekebutikken for å kjøpe seg en fotball.
Han får litt penger av foreldrene sine. Han er ganske liten og vet
verken hva de forskjellige fotballene koster eller hvor mye han har.
Han går med en tilfeldig fotball og pengene til kassen.

Mannen i kassa ser at Einar har fått (Skriv inn antall kroner): 240
Ballen har en pris på (Skriv inn antall kroner): 215
Einar har råd til fotballen og har 25 kroner igjen.



```

# e) -----
print("""Lille Einar skal på lekebutikken for å kjøpe seg en fotball.
Han får litt penger av foreldrene sine. Han er ganske liten og vet
verken hva de forskjellige fotballene koster eller hvor mye han har.
Han går med en tilfeldig fotball og pengene til kassen.
""")

# a) -----
#penger = 200
#pris = 350

# f)
penger = input("Mannen i kassa ser at Einar har fått (Skriv inn antall kroner): ")
penger = int(penger)
# Alternativt:
#penger = int(input("Mannen i kassa ser at Einar har fått (Skriv inn antall kroner): "))

# g)
pris = int(input("Ballen har en pris på (Skriv inn antall kroner): "))

# b) -----
if penger > pris:
    print("Einar har råd til fotballen og har", penger - pris, "kroner igjen.")
# c) -----
else:
    print("Einar mangler", pris - penger, "kroner for å ha råd til fotballen")

```

Oppgave 19 (funksjon_test.py)

Program:

```

def testfunksjon():
    print("test")

testfunksjon()

print("ferdig")

```

Utskrift (Output):

```

test
ferdig

```

Oppgave 20 (funksjon_test2.py)

Program:

```

def gladmelding():
    resultat = "Du kan allerede masse programmering!"
    return "Du er superflink"

print(gladmelding())

```

Utskrift (Output):

Du er superflink

Oppgave 21 (funksjon.py)

Program:

```

def femplusto():
    resultat = 5+2
    return resultat

print(femplusto())

```

Utskrift (Output):

7



Oppgave 22 (funksjon_argument.py)

Program:

```
def x_pluss_3(x):  
    resultat = x + 3  
    return resultat  
  
print(x_pluss_3(2))
```

Utskrift (Output):

5

Oppgave 23 (flere_argumenter.py)

Program:

```
def x_ganger_y(x,y,z):  
    return x * y * z  
  
print(x_ganger_y(5,10,2))
```

Utskrift (Output):

100

Oppgave 24 (valgfri_funk.py)

Program:

```
def tilfeldig_funksjonsnavn(a,b,c):  
    return "hei"  
  
print(tilfeldig_funksjonsnavn(2, 3, 4))
```

Utskrift (Output):

hei

Lag din egen funksjon. Den trenger ikke å gi så mye mening, eller kanskje du vil at den skal gi masse mening? Kjør på, lag noe du har lyst til å lage!

Oppgave 25 (liste.py)

Program:

```
# a) -----  
liste_1 = [1,2,3,4,5]  
liste_2 = [6,7,8,9]  
# b) -----  
print(liste_1[2])  
print(liste_2[-2])  
# c) -----  
print(liste_1[0]+liste_2[3])  
# d) -----  
print(len(liste_1))  
# e) -----  
print(len(liste_2))
```

Utskrift (Output):

3
8
10
5
4



Oppgave 26 (listeaddisjon.py)

Program:

```
# a) -----
x = ["Lister", "er"]
y = ["gøy"]

# b) -----
z = x + y          # z er nå lik ["Lister", "er", "gøy"]
print(z)

# c) -----
print(z[0])
print(z[-3])

# d) -----
print(3*y)

# I lister kan vi bruke pluss og ganger, men ikke minus og dele.
```

Utskrift (Output):

```
['Lister', 'er', 'gøy']
Lister
Lister
['gøy', 'gøy', 'gøy']
```

Oppgave 27 (listception.py)

Program:

```
# a) -----

liste_1 = [2, "fire", 3.14]

# b) -----

liste_1.append(5)
print(liste_1)

# c) -----

liste_2 = ["Enda", "en", "liste"]

# d) -----

liste_2.append(liste_1)
print(liste_2)

# e) -----

print(liste_2[-1])
```

Utskrift (Output):

```
[2, 'fire', 3.14, 5]
['Enda', 'en', 'liste', [2, 'fire', 3.14, 5]]
[2, 'fire', 3.14, 5]
```

Oppgave 28 (while_kopi.py)

Program:

```
x = 1          # Vi setter en startverdi for x

while x <= 4:  # Betingelsen må være oppfylt for at løkken skal kjøre
    print("Nå er x =", x) # Løkkens formål, operasjonen som skal gjentas
    x = x+1        # Vi øker x verdien med 1.

print("While løkken er nå utført")
print("Når er x =", x)
```

Utskrift (Output):

```
Nå er x = 1
Nå er x = 2
Nå er x = 3
Nå er x = 4
While løkken er nå utført
Når er x = 5
```



Oppgave 29 (while.py)

Program:

```
a = 0

while a < 7:
    print("Dette er gøy")
    a = a + 2
```

Utskrift (Output):

```
Dette er gøy
Dette er gøy
Dette er gøy
Dette er gøy
```

Oppgave 30 (while_evig.py)

Program:

a)

```
a = 2

while a < 5:
    print("Uendelig!")
```

b)

```
a = 2

while a < 5:
    print("Uendelig!")
    a = a + 1
```

Utskrift (Output):

a)

```
Uendelig!
Uendelig!
Uendelig!
Uendelig!
Uendelig!Traceback (most recent call last):
  File "/Applications/Spyder.app/Contents/Resc
py3compat.py:356 in compat_exec
    exec(code, globals, locals)
  File ~/Python oppgaver løsning/Python 21-30
print("Uendelig!")
```

```
File /Applications/Spyder.app/Contents/Resc
iostream.py:631 in write
    if self.pub_thread is None:
```

KeyboardInterrupt

b)

```
Uendelig!
Uendelig!
Uendelig!
```

Oppgave 31 (dobling_kopi.py)

Program:

a)

```
x = int(input("Skriv inn ett lite tall: "))
y = int(input("Skriv inn ett stort tall: "))

x_start = x
n = 0

while x < y:
    x = 2 * x
    n = n + 1

print("")
print("Det tar", n, "doblinger før", x_start, "blir større enn eller lik", y)
```

Utskrift (Output):

a) (Som i eksempel)

Skriv inn ett lite tall: 3

Skriv inn ett stort tall: 9

Det tar 2 doblinger før 3 blir større enn eller lik 9



b)

```
x = int(input("Skriv inn ett lite heltall: "))
y = int(input("Skriv inn ett stort heltall: "))
x_start = x
n = 0 # Vi bruker denne variablen til å telle antall triplinger
while x < y:
    x = 3 * x
    n = n + 1 # Antall triplinger øker med en hver gang løkken kjøres.

print("")
print("Det tar", n, "triplinger før", x_start, "blir større enn eller lik", y)
```

c)

```
x = int(input("Skriv inn ett lite heltall: "))
y = int(input("Skriv inn ett stort heltall: "))
x_start = x
n = 0
# c)
y_start = y
while y > x:
    y = 0.5*y
    n += 1
print("Det tar", n, "halveringer før", y_start, "blir mindre enn eller lik", x)
```

b)

```
Skriv inn ett lite heltall: 4
Skriv inn ett stort heltall: 20

Det tar 2 triplinger før 4 blir større enn eller lik 20
```

c)

```
Skriv inn ett lite heltall: 3
Skriv inn ett stort heltall: 100
Det tar 6 halveringer før 100 blir mindre enn eller lik 3
```

Oppgave 32 (while_epsilon.py)

Program:

```
# a) -----
x = 0
epsilon = 1
def f(x):
    return -5*x+302
# b) -----
while f(x) >= 0:
    x = x + epsilon
# c) -----
print("Funksjonen har et nullpunkt mellom x = ", round(x-epsilon,6), "og x =", round(x,6))
```

Utskrift (Output):

c)

Funksjonen har et nullpunkt mellom x = 60 og x = 61

d)

epsilon = 0.1

Funksjonen har et nullpunkt mellom x = 60.3 og x = 60.4

epsilon = 0.01

Funksjonen har et nullpunkt mellom x = 60.4 og x = 60.41

epsilon = 0.001

Funksjonen har et nullpunkt mellom x = 60.4 og x = 60.401

Oppgave 33 (for_kopi.py)

Program:

a)

```
for element in [2, "Sokk", 7]:
    print(element)
```

b) og c)

```
for tak in [2, "Sokk", 7, 5.77]:
    print(tak)
```

a)

```
2
Sokk
7
```

b) og c)

```
2
Sokk
7
5.77
```



Oppgave 34 (priser.py)

Program:

```
x = 1
samlet_pris = 0
priser = [34, 80, 20, 19, 128]
for pris in priser:
    print("Varenummer:", x, "koster", pris, "kroner")
    samlet_pris = samlet_pris + pris
    x = x + 1
print("Samlet koster varene:", samlet_pris, "kroner")
```

Utskrift (Output):

```
Varenummer: 1 koster 34 kroner
Varenummer: 2 koster 80 kroner
Varenummer: 3 koster 20 kroner
Varenummer: 4 koster 19 kroner
Varenummer: 5 koster 128 kroner
Samlet koster varene: 281 kroner
```

Oppgave 35 (for_funk.py)

Program:

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

x = [0, 1, 2, 3, 4]
y = []

for i in x:
    y.append(f(i))

print(y)
```

Utskrift (Output):

```
[0, 1, 4, 9, 16]
```

Oppgave 36 (range.py)

Program:

```
# a) -----
print("a")
r_a = range(5)          # Tilsvare listen [0,1,2,3,4]
print(r_a[0], r_a[1], r_a[2], r_a[-1])

# b) -----
print("b")
r_b = range(3, 10)       # Tilsvare listen [3,4,5,6,7,8,9]
print(r_b[0], r_b[1], r_b[2], r_b[-1])

# c) -----
print("c")
r_c = range(4, 20, 3)     # Tilsvare listen [4,7,10,13,16,19]
print(r_c[0], r_c[1], r_c[2], r_c[-1])
```

Utskrift (Output):

```
a) 0 1 2 3 4
b) 3 4 5 6 7 8 9
c) 4 7 10 13 16 19
```



Oppgave 37 (varer_priser.py)

Program:

```
# a) -----
vare = ["Eple", "Brød", "Skinke", "Paprika"] # Paprika
pris = [10,20,25,12] # 12
# b) -----
for i in range(len(vare)):
    print(vare[i], "koster", pris[i], "kroner")
```

Utskrift (Output):

```
Eple koster 10 kroner
Brød koster 20 kroner
Skinke koster 25 kroner
Paprika koster 12 kroner
```

Oppgave 38 (for_range_funk.py)

Program:

```
# a) -----
def f(x):
    return -x**2+2*x

x = []
y = []

# b) -----
for i in range(10,101,10):
    x.append(i)
    y.append(f(i))

# d) -----
for i in range(len(x)):
    print("(" + x[i] + "," + y[i] + ")")
```

Utskrift (Output):

```
( 10 , -80 )
( 20 , -360 )
( 30 , -840 )
( 40 , -1520 )
( 50 , -2400 )
( 60 , -3480 )
( 70 , -4760 )
( 80 , -6240 )
( 90 , -7920 )
( 100 , -9800 )
```

Oppgave 39 (konsolloppgave)

Program:

```
In [1]: x = 8

In [2]: x<6 and x<10
Out[2]: False

In [3]: x<6 or x<10
Out[3]: True
```

Utskrift (Output):



Oppgave 40 (x_and_or_y.py)

Program:

```
a)
# a) -----
x = 3
y = 5
# b) -----
if x > 4 and y > 4:          # d) or
    print("Begge tallene er større enn 4")  # d) Minst et
# c) -----
else:
    print("Minst ett av tallene er mindre enn eller lik 4")

b)
# a) -----
x = 3
y = 5
# b) -----
if x > 4 or y > 4:          # d) or
    print("Minst ett av tallene er større enn 4")
# c) -----
else:
    print("Begge tallene er mindre enn eller lik 4")
```

Utskrift (Output):

```
c)
Minst ett av tallene er mindre enn eller lik 4
d)
Minst ett av tallene er større enn 4
(Prøv med andre tall på egen hånd)
```

Oppgave 41 (kino.py)

Program:

```
Elise = 14
Marie = 18

if Elise >= 15 and Marie >= 15:
    print("Begge får sett filmen")
else:
    print("De kom dessverre ikke inn.")

Bonus:
Elise = 14
#
Marie = 18
#
if Elise >= 15 and Marie >= 15:
    print("Begge får sett filmen")
#
else:
    if Elise >= 12 and Marie >= 18:
        print("De kom kun inn fordi Marie er voksen")
    else:
        print("De kom dessverre ikke inn.")
..
```

Utskrift (Output):

```
De kom dessverre ikke inn.
Bonus:
De kom kun inn fordi Marie er voksen
```



Oppgave 42 (elif_kopi.py)

Program:

```
x = 7

if x > 10:      # dersom "if-koden" blir kjørt blir ikke elif eller else kjørt.
    print("x er større enn 10!")
elif x > 8:     # Her kunne vi også ha skrevet kun elif x > 8:
    # Dette er fordi elif kommandoen under if kun utløses dersom if ikke kjøres
    print("x er imellom 8 og 10")
else:
    print("x er mindre enn 8")
```

Utskrift (Output):

x er mindre enn 8

Oppgave 43 (karakter_elif.py)

Program:

a) b) c)

x = 4

```
# x = input("Skriv inn en karakter: ")
# x = int(x)
```

```
if x >= 5:
    print("Måloppnåelsen er høy")
elif 3 <= x < 5:
    print("Måloppnåelsen er middels")
else:
    print("Måloppnåelsen er lav")
```

d)

```
x = 4

x = input("Skriv inn en karakter: ")
x = int(x)

if x >= 5:
    print("Måloppnåelsen er høy")
elif 3 <= x < 5:
    print("Måloppnåelsen er middels")
else:
    print("Måloppnåelsen er lav")
```

e)

```
x = input("Skriv inn en karakter: ")
x = int(x)

while x >= 1 and x <= 6:
    if x >= 5:
        print("Måloppnåelsen er høy")
    elif 3 <= x < 5:
        print("Måloppnåelsen er middels")
    else:
        print("Måloppnåelsen er lav")
    x = input("Skriv inn en karakter: ")
    x = int(x)
```

Utskrift (Output):

a) b) c)

Måloppnåelsen er middels

d)

Skriv inn en karakter: 2

Måloppnåelsen er lav

e)

Skriv inn en karakter: 1

Måloppnåelsen er lav

Skriv inn en karakter: 2

Måloppnåelsen er lav

Skriv inn en karakter: 3

Måloppnåelsen er middels

Skriv inn en karakter: 4

Måloppnåelsen er middels

Skriv inn en karakter: 5

Måloppnåelsen er høy

Skriv inn en karakter: 6

Måloppnåelsen er høy

Skriv inn en karakter: 7

Her brøt jeg ut av løkken ved å skrive ett tall større enn 6.



Oppgave 44 (konsolloppgave)

Program:

```
In [1]: from random import *

In [2]: randint(1,6)
Out[2]: 5

In [3]: randint(0,1)
Out[3]: 1

In [4]: round(uniform(0,1),2)
Out[4]: 0.53

In [5]: choice(["Susanne", "Tormod", "Ricardo"])
Out[5]: 'Tormod'

In [6]: choice(["kron", "mynt"])
Out[6]: 'kron'

In [7]: choice([1,2,3,4,5,6])
Out[7]: 4
```

Oppgave 45 (random_kopi.py)

Program:

a)

```
from random import *

Sara_1 = randint(1,6)      # Vi simulerer terningskast og lagrer det.
Sara_2 = randint(1,6)
Lars_1 = randint(1,6)
Lars_2 = randint(1,6)

Sum_Sara = Sara_1 + Sara_2 # Vi finner summen av terningkastene.
Sum_Lars = Lars_1 + Lars_2

if Sum_Lars > Sum_Sara:
    print("Lars vant! Han fikk til sammen:", Sum_Lars, "Sara fikk:", Sum_Sara)
elif Sum_Lars == Sum_Sara:
    print("De fikk", Sum_Lars, "begge to! De fikk like mye.")
else:
    print("Sara vant! Hun fikk til sammen:", Sum_Sara, "Lars fikk:", Sum_Lars)
```

b)

```
from random import *

Sara_1 = randint(1,100) # randint(1,100) # round(uniform(0,100),2)
Sara_2 = randint(1,100) # randint(1,100) # round(uniform(0,100),2)
Lars_1 = randint(1,100) # randint(1,100) # round(uniform(0,100),2)
Lars_2 = randint(1,100) # randint(1,100) # round(uniform(0,100),2)

Sum_Sara = Sara_1 + Sara_2
Sum_Lars = Lars_1 + Lars_2

if Sum_Lars > Sum_Sara:
    print("Lars vant! Han fikk til sammen:", Sum_Lars, "Sara fikk:", Sum_Sara)
elif Sum_Lars == Sum_Sara:
    print("De fikk", Sum_Lars, "begge to! De fikk like mye.")
else:
    print("Sara vant! Hun fikk til sammen:", Sum_Sara, "Lars fikk:", Sum_Lars)
```

c)

```
from random import *

Sara_1 = round(uniform(0,100),2) # randint(1,100) # round(uniform(0,100),2)
Sara_2 = round(uniform(0,100),2) # randint(1,100) # round(uniform(0,100),2)
Lars_1 = round(uniform(0,100),2) # randint(1,100) # round(uniform(0,100),2)
Lars_2 = round(uniform(0,100),2) # randint(1,100) # round(uniform(0,100),2)

Sum_Sara = Sara_1 + Sara_2
Sum_Lars = Lars_1 + Lars_2

if Sum_Lars > Sum_Sara:
    print("Lars vant! Han fikk til sammen:", Sum_Lars, "Sara fikk:", Sum_Sara)
elif Sum_Lars == Sum_Sara:
    print("De fikk", Sum_Lars, "begge to! De fikk like mye.")
else:
    print("Sara vant! Hun fikk til sammen:", Sum_Sara, "Lars fikk:", Sum_Lars)
```

Utskrift (Output):

a)

```
Sara Vant! Hun fikk til sammen: 12 Lars fikk: 3
Lars Vant! Han fikk til sammen: 8 Sara fikk: 5
```

(Samme som i eksempelet, du kan få andre svar)

b)

```
Lars vant! Han fikk til sammen: 129 Sara fikk: 74
```

Du kan få (og vil få) andre svar

c)

```
Lars vant! Han fikk til sammen: 187.69 Sara fikk: 93.11999999999999
```

Du kan få (og vil få) andre svar



Oppgave 46 (random_kast.py)

Program:

```
from random import randint
terning_1 = randint(1,6)
print("På første terningkast fikk jeg:", terning_1)
terning_2 = randint(1,6)
if terning_1 >= terning_2:
    print("Det høysete kastet ga:", terning_1)
else:
    print("Det høysete kastet ga:", terning_2)
```

Utskrift (Output):

På første terningkast fikk jeg: 3
Det høysete kastet ga: 4
(Du kan og vil få andre resultater)

Oppgave 47 (random_alias.py)

Program:

```
from random import randint, uniform as uni
x = randint(0,1)
if x == 0:
    print("Mynt")
else:
    print("Kron")
y = round(uni(0,100),2)
print(y)
```

Utskrift (Output):

Mynt
35.64

(Du kan og vil få andre resultater)

Oppgave 48 (random_import.py)

Program:

```
# a)
import random as rnd

# b)
print(rnd.randint(3,10)) # rnd.uniform

# c)
print(rnd.uniform(2,4)) # rnd.uniform
```

Utskrift (Output):

9
3.599772607718763

(Du kan og vil få andre resultater)

