

Løsningsforslag konte-eksamen DAT120 vinter 2022

Oppgave 1 Basis Python script

```
Flyttall = float(input("Skriv inn et flyttall: "))  
Resultat = flyttall**2  
Print(resultat)
```

Merk: `pow(flyttall, 2)` kan fint erstatte `flyttall**2`.

Oppgave 2 Aritmetisk tallfølge

Opprinnelig løsning

```
a0 = int(input("Skriv inn det første talle, a0: "))  
d = int(input("Skriv inn avstanden mellom tallene i tallfølgen: "))  
n = int(input("Skriv inn antall tall i tallfølgen: "))  
  
tallet = a0  
for i in range(n):  
    print(tallet)  
    tallet += d
```

Alternativ løsning, full score

```
a0 = int(input("Skriv inn det første talle, a0: "))  
d = int(input("Skriv inn avstanden mellom tallene i tallfølgen: "))  
n = int(input("Skriv inn antall tall i tallfølgen: "))  
  
for i in range(a0, a0+(n*d), d):  
    print(i)
```

Oppgave 3 Kodeforståelse, løkker

0:

1: #

2: ##

3: ###

4: ####

5: #####

Oppgave 4 flervalgsoppgave

1. 3

2. **Resultat.** Forklaring: Den globale variabelen resultat settes lik «Resultat» i linje 4. Funksjonen slaar_sammen blir kalt i linje 8. Den ser ut som den setter resultat lik $a + b^2$, men det den gjør er å lage en lokal variabel resultat som skjuler den globale variabelen og sette denne nye lokale variabelen. Den lokale variabelen forsvinner når funksjonen returnerer.

3. 1; 4; 7; 10; 23; 26;

4. **5533.** Forklaring: lengde og bredde blir aldri konvertert til tall, så Python regner dem som strenger. Streng*heltall blir strengen duplisert heltall antall ganger.

5. **Du får en ValueError i linje 4 og programmet stopper der**

6. **Kode 5 er korrekt.** Feilene i de andre kodene:

- 1) Konverterer ikke brukerinput til en float
- 2) Skriver ikke ut resultatet
- 3) Feil i regnestykket, ganger med to i stedet for å opphøye
- 4) Mangler en parentes på slutten av linje 3
- 5) Riktig
- 6) Prøver å skrive ut volumet før det er beregnet

Oppgave 5 Funksjon

```
import turtle
```

```
def taggete_linje(antall_tagger, segment_lengde=15):
```

```
    turtle.left(45)
```

```
    for i in range(antall_tagger):
```

```
        turtle.forward(segment_lengde)
```

```
        turtle.right(90)
```

```
        turtle.forward(segment_lengde)
```

```
        turtle.left(90)
```

```
    turtle.right(45)
```

```
if __name__ == "__main__":
```

```
    taggete_linje(10, 30)
```

```

dummy = input("Skriv inn linjeskift")

turtle.right(90)

taggete_linje(20)

turtle.right(90)

taggete_linje(5, 60)

turtle.right(90)

taggete_linje(20)

turtle.right(90)

turtle.done()

```

Oppgave 6 Problemløsning

Løsning 1: Leit først etter antall 6-ere. Er det 3 eller mer, returner 18. Gjør det samme for 5-ere, 4-ere osv.

```

def tre_like(liste):

    for verdi in range(6, 0, -1):

        teller = 0

        for tall in liste:

            if tall == verdi:

                teller += 1

        if teller >= 3:

            return verdi*3

    return 0

```

Løsning 2: Sorter lista og sjekk deretter på slutten av lista om det er tre like. Gå deretter bakover i lista helt til man når starten. Returner hvis man finner tre like etter hverandre.

```

def tre_like_sortert(liste):

    liste.sort()

    index = len(liste)-1

    while index >= 2:

        if liste[index] == liste[index-1] == liste[index-2]:

            return 3*liste[index]

        index -= 1

    return 0

```

Merk: Begge disse løsningene skal gi full score.

Oppgave 7: Lesing av fil

Deloppgave 1, 2 og 5

try:

```
with open("vaerdata_til_kont_vinter_2022.csv", encoding="UTF8") as fila:
    fila.readline() # LEser inn kolonnetitlene, som ikke skal brukes videre
    total_nedbor = 0.0
    hoyeste_middeltemperatur = -100    # Absurd lavt, alle bør være høyere enn dette
    hoyeste_middeltemperatur_dato = None
    laveste_middeltemperatur = 100     # Absurd høyt, alle bør være lavere enn dette
    laveste_middeltemperatur_dato = None
    for linje in fila:
        linje = linje.replace(",", ".").strip()
        kolonner = linje.split(";")

        if len(kolonner) < 5:    # Håndterer feil antall kolonner
            continue

        try:
            total_nedbor += float(kolonner[4])
            temperatur = float(kolonner[3])

            if temperatur > hoyeste_middeltemperatur:
                hoyeste_middeltemperatur = temperatur
                hoyeste_middeltemperatur_dato = kolonner[2]

            if temperatur < laveste_middeltemperatur:
                laveste_middeltemperatur = temperatur
                laveste_middeltemperatur_dato = kolonner[2]

        except ValueError:    # Håndterer nedbør og temperatur som ikke kan gjøres om til tall
            continue

    print(f"Total nedbør: {total_nedbor}")
    print(f"Høyeste middeltemperatur var {hoyeste_middeltemperatur} grader og var " +
          f"på datoen {hoyeste_middeltemperatur_dato}")
```

```

        print(f"Laveste middeltemperatur var {laveste_middeltemperatur} grader og var " +
              f"på datoen {laveste_middeltemperatur_dato}")
except FileNotFoundError:
    print("Finner ikke denne fila!")
except:
    print("Feil under lesing av fila!") # Håndterer feil under lesing

```

Deloppgave 3

```

with open("vaerdata_flere_stasjoner.csv", encoding="UTF8") as fila:
    sokestreng = input("Skriv del av navn på værstasjon: ")
    fila.readline() # LEser inn kolonnetitlene, som ikke skal brukes videre
    total_nedbor = 0.0
    hoyeste_middeltemperatur = -100 # Absurd lavt, alle bør være høyere enn dette
    hoyeste_middeltemperatur_dato = None
    laveste_middeltemperatur = 100 # Absurd høyt, alle bør være lavere enn dette
    laveste_middeltemperatur_dato = None
    for linje in fila:
        linje = linje.replace(",", ".").strip()
        kolonner = linje.split(";")

        if len(kolonner) < 5: # Håndterer feil antall kolonner
            continue

        if kolonner[0].find(sokestreng) == -1:
            continue

        try:
            total_nedbor += float(kolonner[4])
            temperatur = float(kolonner[3])

            if temperatur > hoyeste_middeltemperatur:
                hoyeste_middeltemperatur = temperatur
                hoyeste_middeltemperatur_dato = kolonner[2]

            if temperatur < laveste_middeltemperatur:

```

```

        laveste_middeltemperatur = temperatur

        laveste_middeltemperatur_dato = kolonner[2]

    except ValueError:    # Håndterer nedbør og temperatur som ikke kan gjøres om til tall
        continue

    print(f"Total nedbør: {total_nedbor}")

    print(f"Høyeste middeltemperatur var {hoyeste_middeltemperatur} grader og var " +
          f"på datoen {hoyeste_middeltemperatur_dato}")

    print(f"Laveste middeltemperatur var {laveste_middeltemperatur} grader og var " +
          f"på datoen {laveste_middeltemperatur_dato}")

```

Deloppgave 4

with open("vaerdata_flere_stasjoner.csv", encoding="UTF8") as fila:

```

    total_nedbor = dict()

    fila.readline() # LEser inn kolonnetitlene, som ikke skal brukes videre

    hoyeste_middeltemperatur = dict()
    hoyeste_middeltemperatur_dato = dict()
    laveste_middeltemperatur = dict()
    laveste_middeltemperatur_dato = dict()

    for linje in fila:

        linje = linje.replace(",", ".").strip()

        kolonner = linje.split(";")

        if len(kolonner) < 5:    # Håndterer feil antall kolonner
            continue

        try:

            if kolonner[0] in total_nedbor:

                total_nedbor[kolonner[0]] += float(kolonner[4])

            else:

                total_nedbor[kolonner[0]] = float(kolonner[4])

            temperatur = float(kolonner[3])

            if kolonner[0] in hoyeste_middeltemperatur:

```

```

        if temperatur > hoyeste_middeltemperatur[kolonner[0]]:
            hoyeste_middeltemperatur[kolonner[0]] = temperatur
            hoyeste_middeltemperatur_dato[kolonner[0]] = kolonner[2]
        else:
            hoyeste_middeltemperatur[kolonner[0]] = temperatur
            hoyeste_middeltemperatur_dato[kolonner[0]] = kolonner[2]

    if kolonner[0] in laveste_middeltemperatur:
        if temperatur < laveste_middeltemperatur[kolonner[0]]:
            laveste_middeltemperatur[kolonner[0]] = temperatur
            laveste_middeltemperatur_dato[kolonner[0]] = kolonner[2]
        else:
            laveste_middeltemperatur[kolonner[0]] = temperatur
            laveste_middeltemperatur_dato[kolonner[0]] = kolonner[2]

except ValueError:    # Håndterer nedbør og temperatur som ikke kan gjøres om til tall
    continue

with open("totaler_for_alle_vaerstasjoner.csv", "w", encoding="utf8") as skrivefil:
    skrivefil.write("Værstasjon; Total nedbør; høyeste middeltemperatur; dato; laveste
middeltemperatur; dato\n")

    for vaerstasjon in total_nedbor:
        skrivefil.write(f"{vaerstasjon}; ")
        skrivefil.write(f"{total_nedbor[vaerstasjon]}; ")

        if vaerstasjon in hoyeste_middeltemperatur:    # Oppgaven spør ikke om denne sjekken,
ikke nødvendig for full score. Bruk av dicts er det viktigste her.
            skrivefil.write(f"{hoyeste_middeltemperatur[vaerstasjon]}; " +
                f"{hoyeste_middeltemperatur_dato[vaerstasjon]}; ")
            skrivefil.write(f"{laveste_middeltemperatur[vaerstasjon]}; " +
                f"{laveste_middeltemperatur_dato[vaerstasjon]}\n")
        else:
            skrivefil.write("; \n")

```

Oppgave 8 objektorientert programmering

def poeng(kortstokk):

sum = 0

if len(kortstokk) == 0: # For å ikke få exception hvis kortstokken er tom. Lite viktig.

return 0

korttype = kortstokk.kortene[0].korttype

for kortet in kortstokk.kortene:

if kortet.korttype != korttype:

return 0

if kortet.verdi == 1:

sum += 15

elif kortet.verdi > 10:

sum += 10

else:

sum += kortet.verdi

return sum