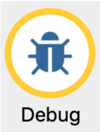
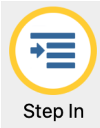
Test deg selv 2

1.1

Forklar programmene nedenfor.

Du kan skrive inn koden i Mu og bruke



og


a.

```
for n in range(50):
    partall = 2*n
    print(partall)
```

b.

```
for n in range(50):
    oddetall = 2*n + 1
    print(oddetall)
```

c.

```
for n in range(0, 101, 2):
    print(n)
```

d.

```
for n in range(1, 100, 2):
    print(n)
```

e.

```
for n in range(1, 100):
    if n%2 == 0:
        print(n)
```

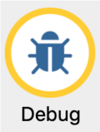
f.

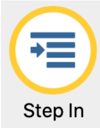
```
for n in range(1, 100):
    if n%2 != 0:
        print(n)
```

1.2

Forklar programmene nedenfor

Du kan skrive inn koden i Mu og bruke



og



```
sum = 0

for n in range(6):
    partall = 2*n
    sum = sum + partall

print(sum)
```

b.

```
sum = 0

for n in range(5):
    oddetall = 2*n + 1
    sum = sum + oddetall

print(sum)
```

2.1 Fliser

Espen har akkurat nok kvadratiske fliser til å dekke et rutenett som måler $n \cdot n$. Hvor mange fliser blir til overs dersom han bruker flisene til å dekke

a et rutenett på $(n+2) \cdot (n-2)$

b et rutenett på $(n+k) \cdot (n-k)$

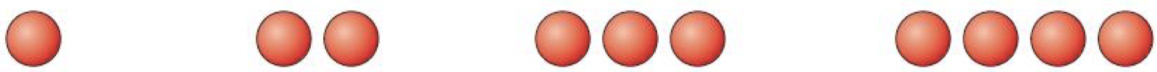
2.2 Løkker

a. Lag et program som skriver ut partallene mellom 1000 og 12000

b. Lag et program som skriver ut oddetallene mellom 500 og 750

NB: Bare lever koden, ikke resultetet 😃

2.3 Figurtall

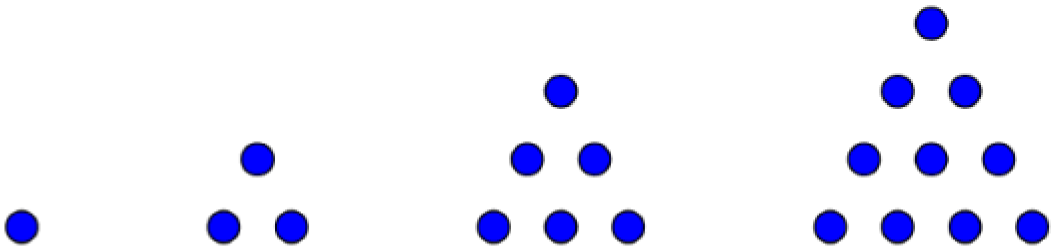


a. Skriv opp de 5 første .tallene

b. Lag et program som skriver ut de 10 første tallene.

c. Bestem summen av de 10 første tallene.

2.4 Trekanttallene



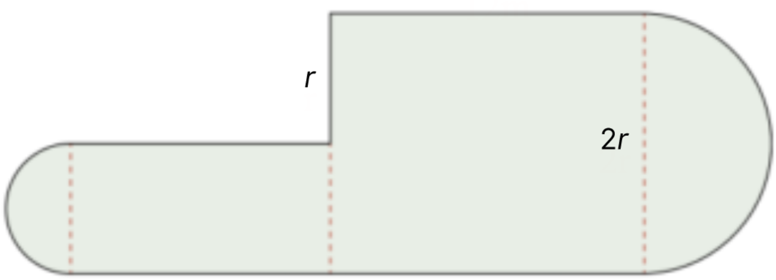
a. Skriv opp de 5 første tallene.

b. Lag et program som skriver ut de 10 første tallene.

c. Bestem summen av de 10 første trekanttallene.

3.1 Gjerde

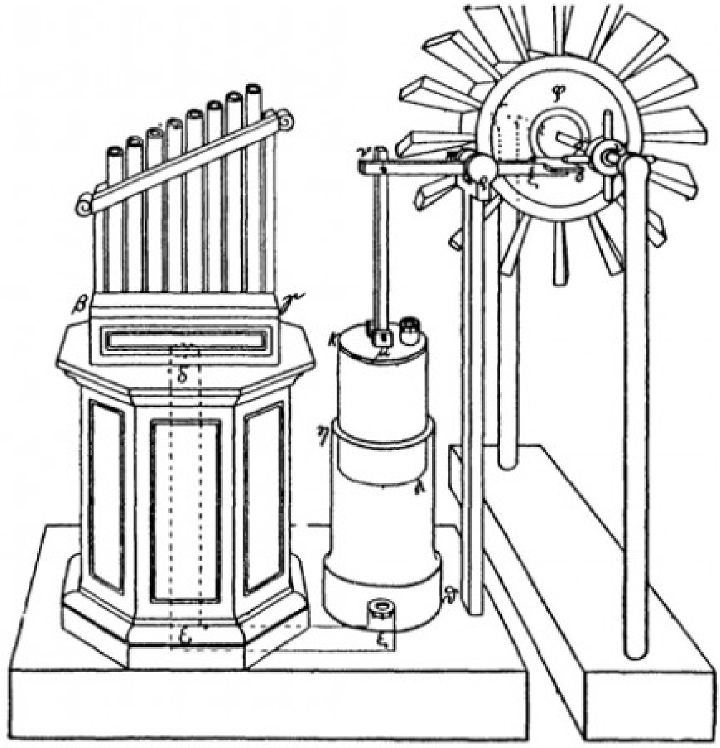
Lavrans skal sette opp gjerde rundt et område. Området består av to halvsirkler og to rektangler slik figuren viser:



a Lag et uttrykk for omkretsen av området.

b Han bruker 200 m gjerde. Finn r.

3.2 Kvadratrottilnærming



Heron fra Alexandria er en kjent matematiker fra antikken. Han laget blant annet en algoritme for å finne tilnærmingsverdier for kvadratroten av et naturlig tall, n:

- Velg et tall, a, i nærheten av det du tror svaret blir
- Regn ut tallet

  $$b = \frac{1}{2}(a + \frac{n}{a})$$

- Gjenta punkt 2 til du får et tall som er så nøyaktig som du ønsker. b blir den nye a-en.

a. Sett a = 1 og n = 4, og utfør algoritmen 3 ganger for hånd. Hva ble svaret?

b. Bruk en for-løkke til å utføre algorithmen i Python for tallet a = 1 og n = 3. Hva er færest antall iterasjoner som trengs og å få en nøyaktighet på 15 desimaler.

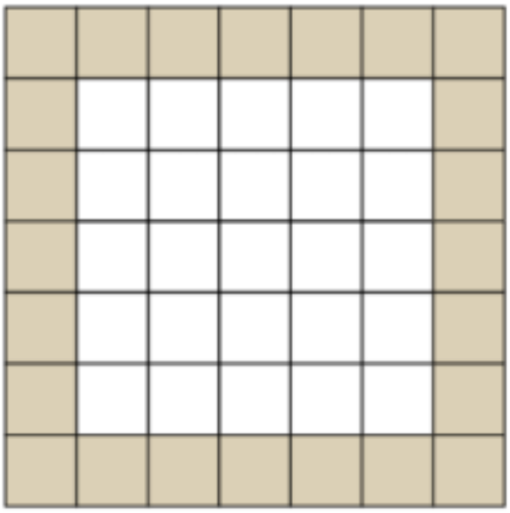
Skriv ut svaret og sammenlign med

```
from math import sqrt
print(sqrt(n))
```

c. Gjør det samme som i **b**, men med a = 1 og n = 3333333. Hvor mange iterasjoner trengs for å oppnå 15 desimaler nøyaktighet nå?

3.3 Steinheller

Et område av hagen til Mehmet er dekt med kvadratiske steinheller. Det er 5 heller i hver retning og derfor 5^2 heller til sammen. Mehmet vil legge en ny rad med heller rundt hele området. Figuren nedenfor viser området på $5 \cdot 5$ heller med den nye raden omkring markert med farge.



a Hvor mange heller trenger Mehmet for å legge den nye raden?

b Finner du flere måter å regne ut svaret på? Forklar hver regnemåte geometrisk.

Et annet område består av $x \cdot x$ heller. Vi legger en ny rad med heller rundt området.

c Hvilke av følgende regnemåter gir oss antall heller vi trenger?

1 $(x+1)^2$

2 $(x+2) \cdot 2 + x \cdot 2$

3 $4 \cdot (x+2)$

4 $4 \cdot (x+1)$

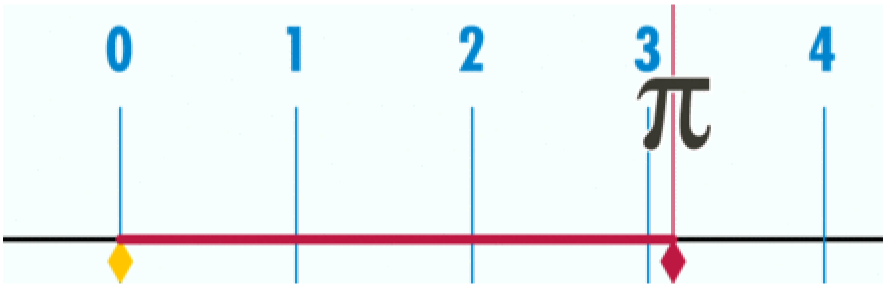
5 $4 \cdot (x+2) - 4$

6 $(x+2)^2 - x^2$

d Hvor mange heller er det i det opprinnelige området hvis vi trenger 52 heller til den nye raden omkring?

3.4 Tilnærming av π

Den matematiske konstanten pi (π) er eit irrasjonalt tal definert som omkrinsen til ein sirkel dividert med diameteren til sirkelen.



Du skal i denne oppgaven bestemme en tilnærmet verdi av π.

Vi skal ta utgangspunkt i Gottfried Leibniz sin metode for å finne 1/4 av omkretsen til en sirkel med radius lik 1, altså π/4.

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots = \frac{\pi}{4}.$$

a. Vi ser at utrykket bytter fortegn for annet hvert ledd.

Bruk egenskapen til $(-1)^n$ og lag et program som skriver ut:

1, -1, 1, -1, 1, -1, 1, -1, 1, -1

Dette kan du nå bruke som tellerene i brøkleddene.

b. Utvid programmet til også å skrive ut nevnerne fra samme for-løkke som **a**:

1, 3, 5, 7, 9, 11, 13, 15, 17, 19

Hint: 1.1 **b**

c. Sett sammen teller og nevner og skriv ut 1, 0.333, 0.2, ...

d. Regn ut summen til tallene i **c** og gang resultatet med 4. Hint: 1.2 **b**.

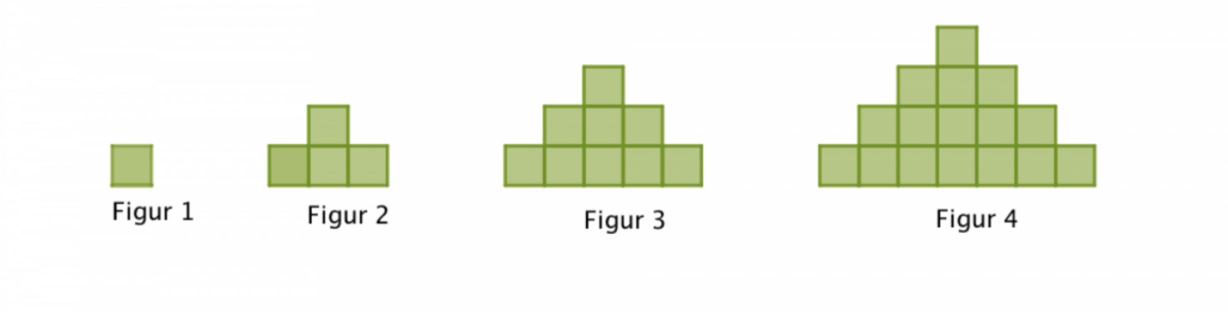
Sammenlign resultatet med

```
from math import pi
print(pi)
```

Øk range til 10 ** 3, 10 ** 4, osv. Hvor mange desimaler klarer du å tilnærme?

Ekstraoppgaver

4.1 Stabel

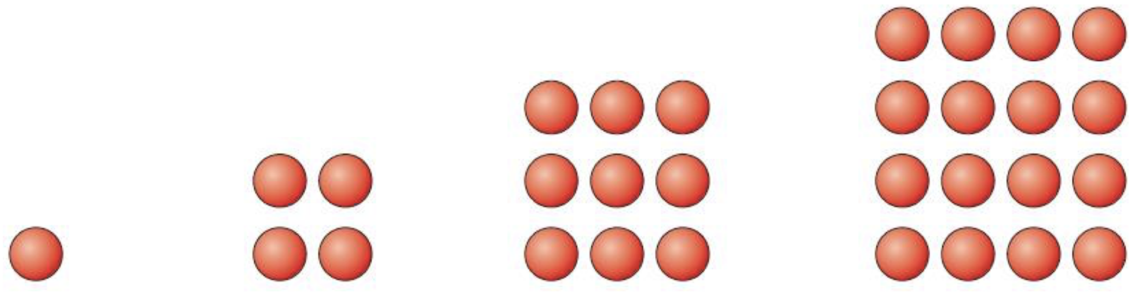


a. Skriv opp de 5 første tallene.

b. Lag et program som skriver ut de 10 første tallene.

c. Bestem summen av de 10 første tallene.

4.2 Kvadrattallene



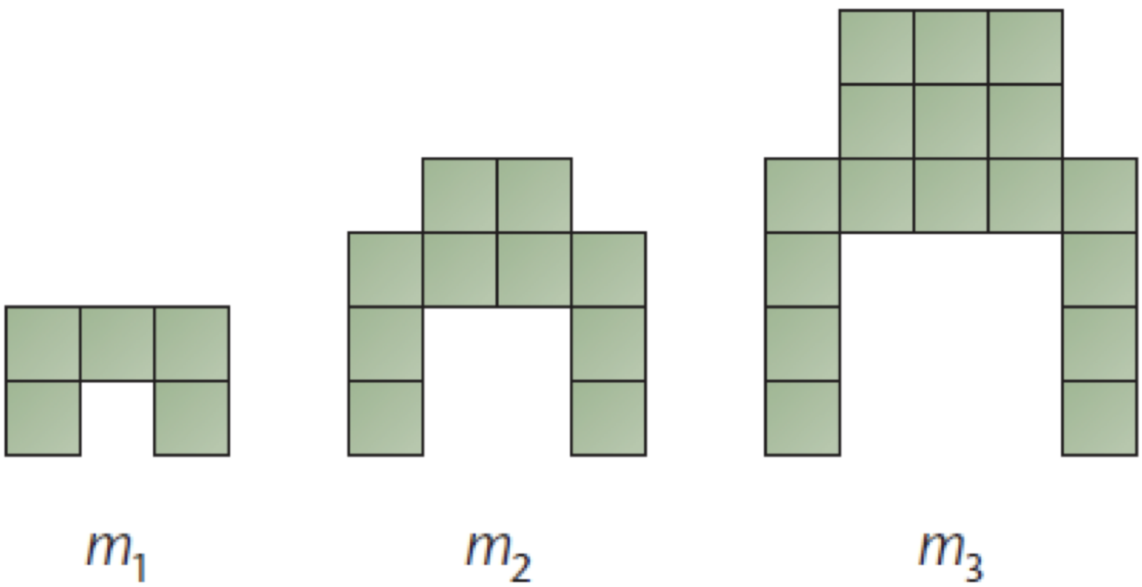
a. Skriv opp de 5 første tallene

b. Lag et program som skriver ut de 10 første tallene.

c. Bestem summen av de 10 første tallene.

d. Sammenlign resultatene fra 4.1. Hvorfor blir det slik?

4.3 En kombinasjon



a. Skriv opp de 5 første tallene

b. Lag et program som skriver ut de 10 første tallene.

c. Bestem summen av de 10 første tallene.