



1 a) Output skal vise: 69

b) En gyldig løsning er:

```
disp(3*(18/(14-2))+4);
```

Svaret blir da 8.5.

2 a)

```
tall1 = 7;  
tall2 = 3;  
tall1 + tall2
```

Gir output: ans = 10

```
tall2 = 4;  
2*(tall1^10)+tall2^8
```

Gir output: ans = 565016034

b)

```
tall3 = 2*(tall1^10) + tall2^8;  
tall3^0.1
```

Gir output: ans = 7.5025

c)

```
tall1 = 10;  
tall1 = tall1 + 1;  
tall1 = tall1 + 1;  
tall1 = tall1 + 1;
```

Variabelen `tall1` skal nå ha verdien 13. Det kan sjekkes med funksjonen `disp` slik:

```
disp(tall1);
```

d) Når man skriver formler som kan gjenbrukes er det lurt å bruke variabler i formelen i stedet for å bruke verdiene direkte. Senere vil du lære å plassere slike setninger i funksjoner, på den måten kan du enkelt bruke den på nytt om du får bruk for den.

```
laan = 1000;  
rente = 0.04;  
laan*((1+rente)^100)
```

Gir output: ans = 5.0505e+04

3 a) `isprime` finner ut hvorvidt et eller flere tall er primtall

b) 131, 79, 127 og 13 er primtall

- c) I sin helhet regner linjen ut hvor mange primtall det finnes mellom 1 og 100.

Da dette er funksjonskall som argument til funksjonskall vil setningen løses fra innerste til ytterste kall. Returneringsverdien til den første funksjonen blir brukt som argument til neste funksjonen osv.

```
1:100
```

Lager en vektor med alle tallene fra 1 til 100.

```
isprime(...)
```

Lager en ny vektor med 100 elementer hvor hvert element er 0 eller 1 avhengige om det korresponderende tallet i den første vektoren er et primtall.

```
sum(...)
```

Summerer alle tallene i vektoren som isprime returnerer.

```
disp(...)
```

Skriver ut summen til skjermen.

4

a)

```
kubeV = kubeVolum(2)
kuleV = kuleVolum(1)
differanse = kubeV-kuleV
```

Som gir at kubens volum er 8, kulens volum er 4.1888 og dermed er kubens volum 3.8112 større enn kulens.

b)

```
function y = pyramideVolum(lengde, hoyde)
    y = 1/3*lengde^2*hoyde;
end
```

Se også vedlagt fil pyramideVolumLF.m