

NTNU

Kunnskap for en bedre verden

TDT4105 Informasjonsteknologi, grunnkurs

Matlab:

**Betinget programutførelse (valg: IF og SWITCH)
og løkker (FOR)**

Rune Sætre og Anders Christensen (satre og anders @idi.ntnu.no)

Læringsmål

- Mer om IF-setningen og betinget programutførelse
 - IF-ELSEIF-ELSE-setningen (kap 4.3)
- SWITCH-setningen
 - Direkte valg mellom flere alternativer (kap 4.4)
- Løkker
 - FOR-setningen (kap 5.1, og kap 5.2 doble løkker)
 - Et bestemt antall gjentakelser
- Kap. 4.1-4.4 og 5.1-5.2 i Matlab-boka (3rd Edition)

Referansegruppemøte M1

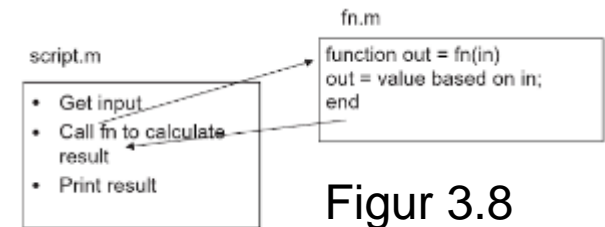
- Alle ref.-gruppefolk, kom frem i pausen så vi kan finne et møtetidspunkt som passer alle i neste uke:
 - MBIOT5 (Bioteknologi), (@stud)
 - MTBYGG (Bygg- og miljøteknikk), (@stud)
 - MTENERG (energi og miljø), (@stud)
 - *MPETR* (Petroleumsfag), (@stud)
 - *MTTEKGEO* (Tekniske Geofag), (@stud)

Referansegruppemøte M2 (Rune)

- Første ref.-gruppemøte blir 28. september (uke 39):
 - BGEOL (geologi), Martin Sesaker (martises@stud)
 - MTING (ing og IKT), Sigurd Sandvoll Sundberg (sigurssu@stud)
 - MTIØT (indØk-Tek, EM, Mar, Prod), Marie Tyssen Bruu (marietb@stud)
 - MTMART (marin), Hans Christian Høyland (hchoylan@stud)
 - MTPROD (produktiv & prod), Torgeir Ketilsønn Kjevik (torgeikk@stud)
- Si ifra til en personene over om ting som er bra eller ikke i løpet av uken
- Er det andre studieretninger til stede i forelesningene?
 - Bli med i referansegruppen!

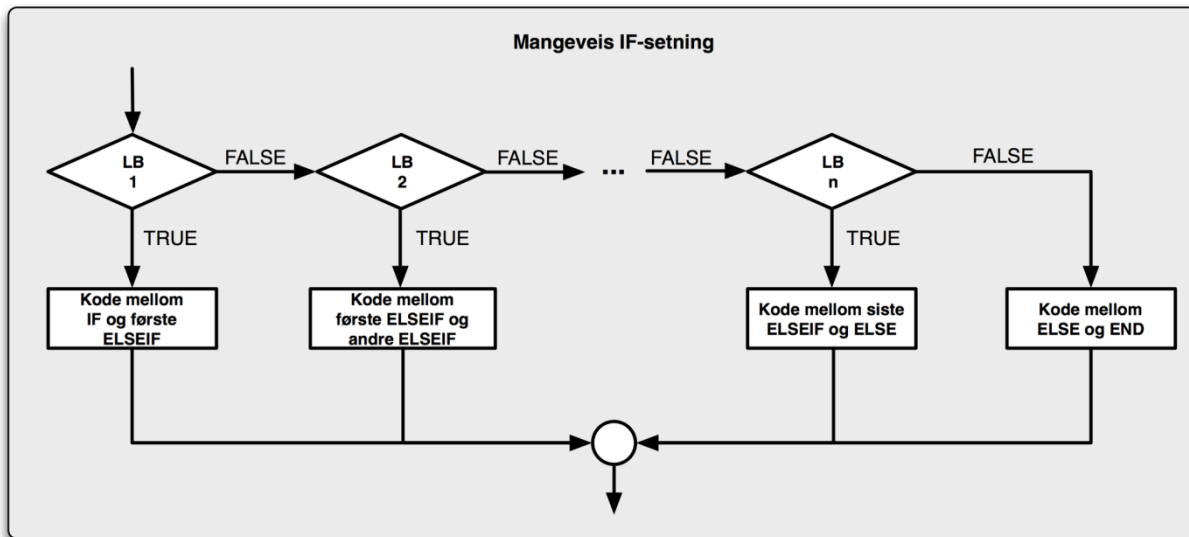
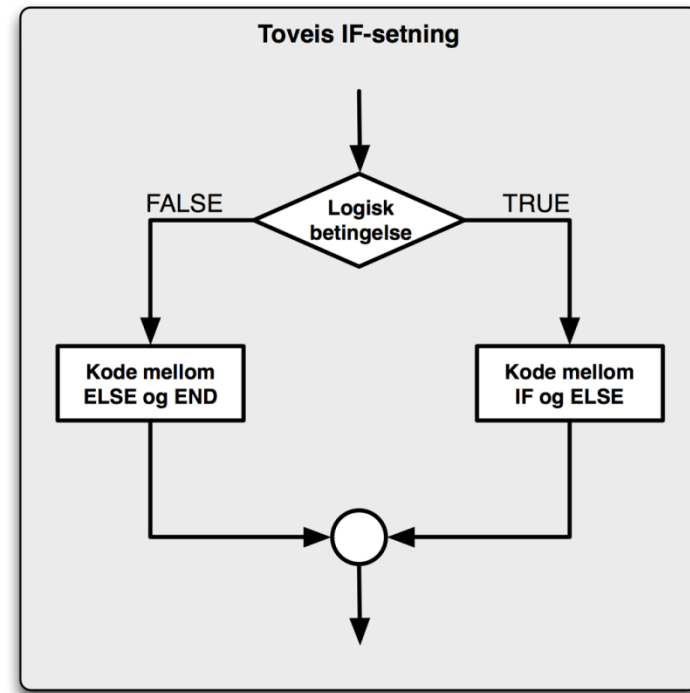
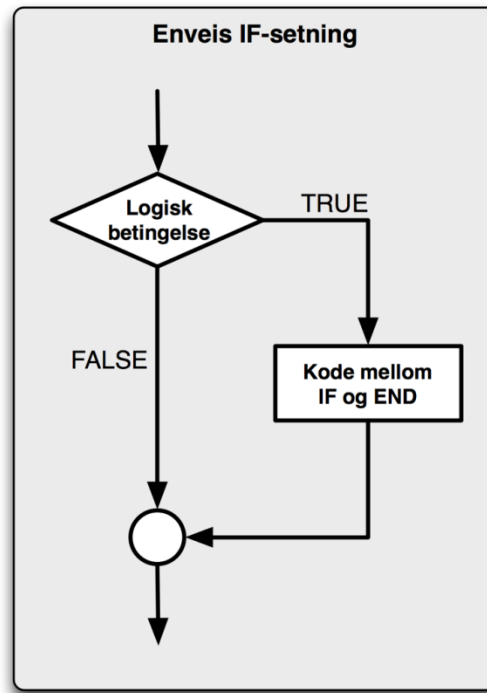
Repetisjon (Kap. 3.7.3.1, side 102): Forskjellen på Program og Funksjon

- Hovedprogram
 - Les inn data (fra tastatur eller fil eller ...)
 - Behandle data, kalle funksjoner
 - Skriv ut resultatet (til skjerm eller fil eller ...)



- Funksjon
 - «function», og definer returvariabel, navn, innparametre i 1. linje
 - Behandle innparametre
 - Lagre svar-verdien i returvariabelen
 - end % function
 - INGEN utskrift (som regel)

IF



Poeng til bokstavkarakter

- 89-100 poeng gir A
- 77-88 poeng gir B
- 65-76 poeng gir C
- 53-64 poeng gir D
- 41-52 poeng gir E
- 0-40 poeng gir F (stryk)
- Minuspoeng og mer enn 100 poeng har udefinert resultat (noe er feil), bør varsles
- Lag et funksjon `poeng2karakter`, som får poeng inn, og returnerer karakter ut

Alt 1: Nøstet IF

```
function karakter = poeng2karakter(poeng)
% poeng -> bokstavkarakter, noestet IF
% validerer input
if (poeng < 0) || (poeng > 100)
    error('Feil inndata')
end

if poeng >= 89
    karakter = 'A';
else
    if poeng >= 77
        karakter = 'B';
    else
        if poeng >= 65
            karakter = 'C';
        else
            if poeng >= 53
                karakter = 'D';
            else
                if poeng >= 41
                    karakter = 'E';
                else
                    karakter = 'F';
                end
            end
        end
    end
end
end
end
end % function
```


Alt 2: ELSEIF

```
function karakter =  
poeng2karakter2(poeng)  
% poeng -> bokstavkarakter, IF-  
ELSEIF  
  
% validerer input  
if poeng < 0 || poeng > 100  
    error('Feil inndata')  
end  
  
if poeng >= 89  
    karakter = 'A';  
elseif poeng >= 77  
    karakter = 'B';  
elseif poeng >= 65  
    karakter = 'C';  
elseif poeng >= 53  
    karakter = 'D';  
elseif poeng >= 41  
    karakter = 'E';  
else  
    karakter = 'F';  
end  
  
end % function
```

Alt 3: En IF pr. intervall

```
function karakter = poeng2karakter3(poeng)
% poeng -> bokstavkarakter, en IF pr poengintervall

% validerer input
if (poeng < 0) || (poeng > 100)
    error('Feil inndata')
end

if poeng >= 89
    karakter = 'A';
end

if poeng >= 77 && poeng < 89
    karakter = 'B';
end

if poeng >= 65 && poeng < 77
    karakter = 'C';
end

if poeng >= 53 && poeng < 65
    karakter = 'D';
end

if poeng >= 41 && poeng < 53
    karakter = 'E';
end

if poeng < 41
    karakter = 'F';
end

end % function
```

Valg mellom alternativene

- Alle tre alternativene virker
- Ved nøstet IF (alt 1) og med ELSEIF (alt 2) er *rekkefølgen* av betingelsene viktig
- Valgkriterier:
 - Enklest å få rett
 - Enklest å lese (forstå)
 - Minst tekst
 - Mest effektiv
- I dette tilfellet vil mange foretrekke løsningen med IF-ELSEIF-ELSE
 - Mest kompakt, enklest å forstå

SWITCH-setningen

```
switch <uttrykk>
  case <verdi> eller {<verdi-1>, ... <verdi-n>}
    <setninger>
  case <verdi> eller {<verdi-1>, ... <verdi-n>}
    <setninger>
  ...
  otherwise
    <setninger>
end
```

- Velger “case” som matcher <uttrykk> og utfører de tilhørende setningene.
- Ingen match -> utfører otherwise-setningene.

Bestem nasjonalitet for bilmerke

```
function nasjonalitet = bilmerke2nasjonalitet(merke)

    switch merke
        case 'Volvo'
            nasjonalitet = 'Svensk';
        case {'VW', 'Audi', 'BMW'}
            nasjonalitet = 'Tysk';
        case {'Toyota', 'Nissan'}
            nasjonalitet = 'Japansk';
        otherwise
            nasjonalitet = 'Ukjent';
    end

end % function
```

```
>> bilmerke2nasjonalitet('Volvo')
ans =
Svensk
>> bilmerke2nasjonalitet('Toyota')
ans =
Japansk
>> bilmerke2nasjonalitet('Mini')
ans =
Ukjent
>>
```

FOR-løkker

```
FOR tellevariabel = <start>:<inkrement>:<sluttverdi>  
    <setninger>  
END
```

- Setningene i løkke-kroppen gjentas en gang for hver verdi av tellevariabelen
- ```
FOR i = 1:1:5
 disp(i*i)
END
```
- Skriver ut de 5 første kvadrattallene.

```
>> for i=1:1:5 disp(i*i); end
 1

 4

 9

 16

 25

>>
```

# Summer heltallene 1..N

```
function sum = sumHeltall(n)
% summerer heltallene fra 1 tom. n

% setter startverdi
sum = 0;

for i = 1:1:n
 sum = sum + i;
end % for

end % function
```

sumHeltall.m

```
>> sumHeltall(1)
ans =
 1
>> sumHeltall(2)
ans =
 3
>> sumHeltall(5)
ans =
 15
>> sumHeltall(100)
ans =
 5050
>>
```

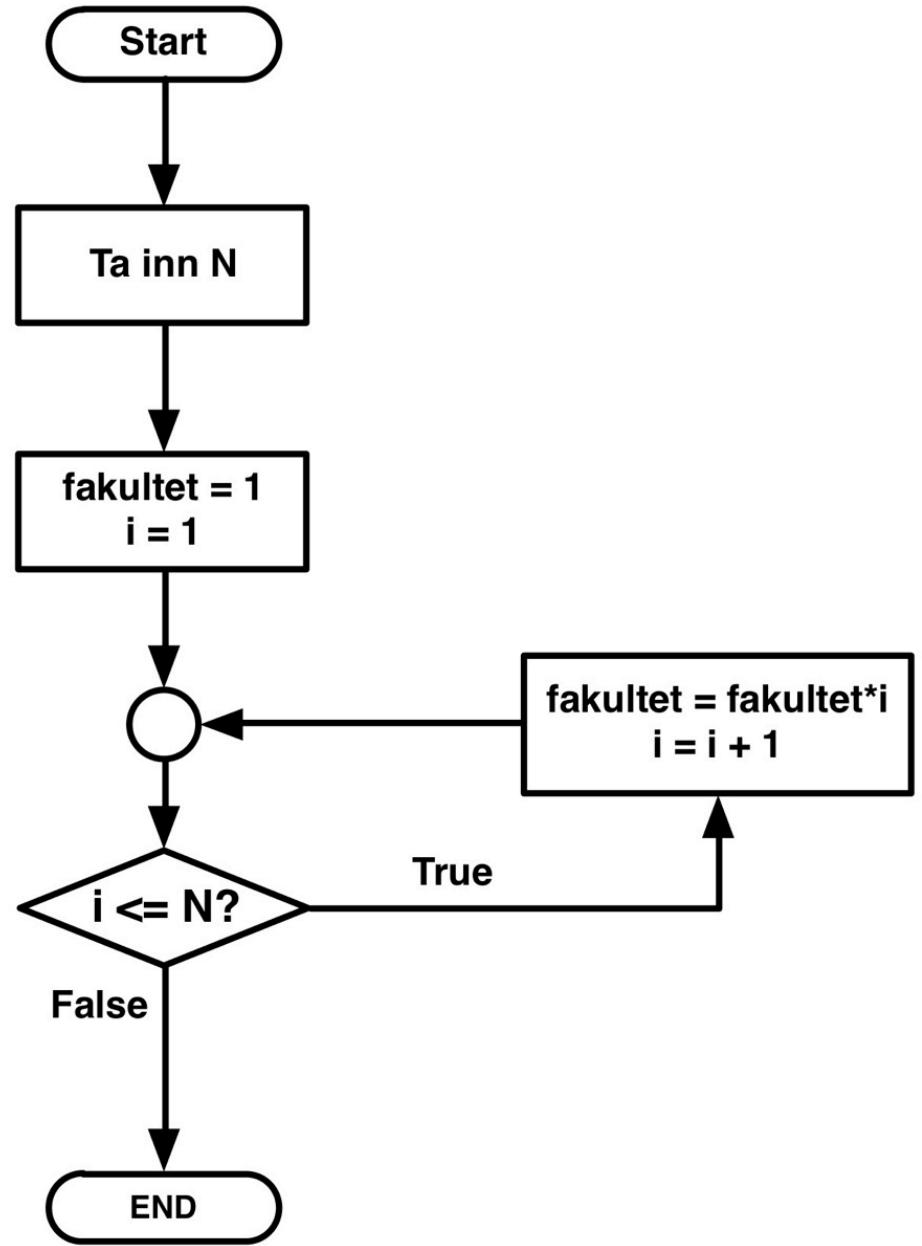
# Fakultetsfunksjonen

- $0! = 1$
- $1! = 1$
- $2! = 1*2 = 2$
- $N! = 1*2*3*...*(n-1)*n$
  
- Lager en fakultetsfunksjon med FOR-løkke





# Flytskjema



# Programkode

fakultet.m

```
function f = fakultet(n)
% beregner n!

if n<0
 error('Feil i fakultet: Negativ innparameter')
end

% startverdi
f = 1;

for i = 1:1:n
 f = f*i;
end % for

end %function
```

```
>> fakultet(0)
ans =
 1
>> fakultet(1)
ans =
 1
>> fakultet(6)
ans =
 720
>> fakultet(-1)
??? Error using ==> fakultet at 5
Feil i fakultet: Negativ innparameter
```

# FOR-løkker kan nøstes

```
% tommer kommandovinduet
clc

for rad=1:1:10
 for kol=1:1:10
 tabell(rad, kol) = rad*kol;
 end
end

disp(tabell);
```

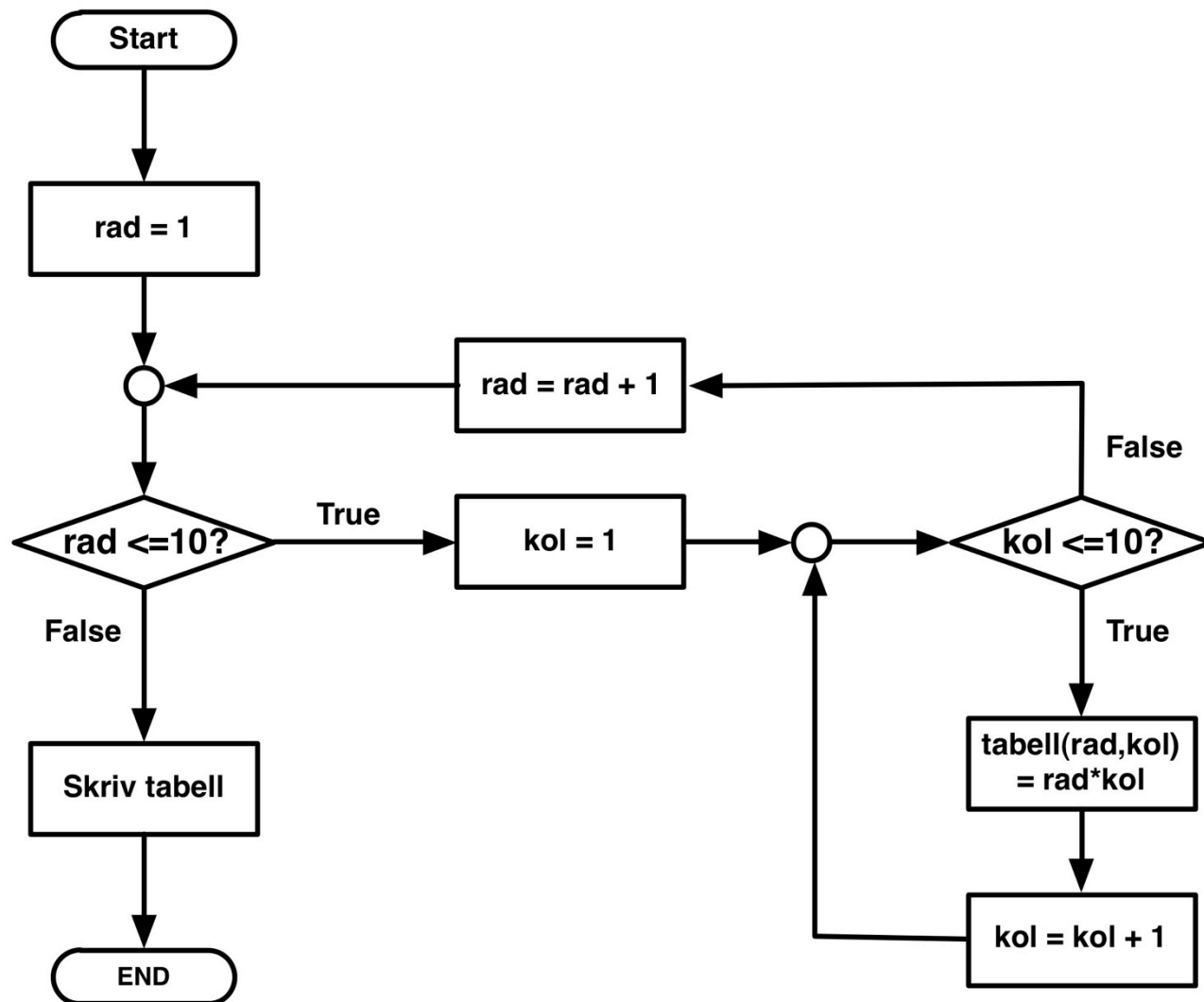
gangetabell.m

# Utskrift

|    |    |    |    |    |    |    |    |    |     |
|----|----|----|----|----|----|----|----|----|-----|
| 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10  |
| 2  | 4  | 6  | 8  | 10 | 12 | 14 | 16 | 18 | 20  |
| 3  | 6  | 9  | 12 | 15 | 18 | 21 | 24 | 27 | 30  |
| 4  | 8  | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40  |
| 5  | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50  |
| 6  | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60  |
| 7  | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70  |
| 8  | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80  |
| 9  | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90  |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

&gt;&gt;

# Flytdiagram



# Telle forekomster av gitt verdi i tabell

|   |   |   |   |
|---|---|---|---|
| 2 | 1 | 3 | 2 |
| 4 | 1 | 1 | 3 |
| 1 | 4 | 4 | 2 |
| 2 | 3 | 3 | 1 |

- Inputt: Tabell og verdi. Returvariabel: antall av v i T
- Finne størrelsen på tabellen (dimensjonene)
  - `size(<tabell>)` returnerer antall rader og kolonner i <tabell>.
  - `[rader, kolonner] = size(<tabell>)`
- Ingen forekomster i utgangspunktet (antall = 0)
- Gå så gjennom tabellen
  - Rad for rad, kolonne for kolonne
  - Øke antall med en for hvert "treff"
- Har resultatet (i antall) når alle elementene er sjekket

```
function antall = antallAvVerdi(tabell, verdi)
% teller antall forekomster av verdi i tabell

% sjekker størrelsen på tabellen
[antRader, antKolonner] = size(tabell);

% startverdi
antall = 0;

% går gjennom tabellen
for rad = 1:antRader
 for kol = 1:antKolonner

 if tabell(rad,kol) == verdi
 antall = antall + 1;
 end
 end
end
end % function
```

```
>> m = randi(3,10,10)
```

```
m =
```

```

 3 2 1 3 3 2 1 1 2 1
 1 2 3 1 1 3 1 3 1 3
 1 3 2 2 3 2 2 1 1 1
 1 3 1 3 1 2 3 3 3 3
 1 1 1 3 3 3 3 1 1 3
 3 2 2 3 2 1 1 2 3 3
 3 2 3 2 1 3 2 1 2 1
 1 2 2 1 1 3 2 2 3 2
 3 3 2 1 2 2 1 3 1 1
 1 3 1 1 2 2 2 2 2 3

```

```
>> antallAvVerdi(m,8)
```

```
ans =
```

```
0
```

```
>> antallAvVerdi(m,1)
```

```
ans =
```

```
36
```

```
>> antallAvVerdi(m,2)
```

```
ans =
```

```
30
```

```
>> antallAvVerdi(m,3)
```

```
ans =
```

```
34
```

```
>>
```



# FOR-løkker, variasjon

```
FOR <tellevariabel> = <start>:<inkrement>:<slutt>
 <setninger>
END
```

- Setningene i løkke-kroppen gjentas en gang for hver verdi av tellevariabelen

```
FOR <tellevariabel> = <tabell>
 <setninger>
END
```

- Løkketkroppen utføres en gang for hvert element i tabell

# Summerer de 10 minste primtallene

```
% Skriptet summerer de 10 minste primtallene

% Tømmer kommandovinduet, fjerner alle variabler
clc, clear

primtall = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29];

sum = 0;

for i = primtall
 sum = sum + i;
 fprintf('Har lagt til %d, summen er nå %d\n', i, sum)
 pause(1) % venter i 1 sekund
end

fprintf('\nSluttsummen ble %d\n', sum)
```

# Programkjøring

```
Har lagt til 2, summen er nå 2
Har lagt til 3, summen er nå 5
Har lagt til 5, summen er nå 10
Har lagt til 7, summen er nå 17
Har lagt til 11, summen er nå 28
Har lagt til 13, summen er nå 41
Har lagt til 17, summen er nå 58
Har lagt til 19, summen er nå 77
Har lagt til 23, summen er nå 100
Har lagt til 29, summen er nå 129
```