```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                   MAT-1051                          %
%           Obligatorisk innlevering "A"              %
%               Automasjon Y-veien                    %
% Fredrik Eilertsen, Morten Bertheussen, Sondre Aspmo %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

VEKT  =  1300;   %Vekt på bilen [kg]
FARTB =  75/3.6; %Fart på bilen [m/s]
FARTA =  2.7;    %Akselerasjon  [m/s^2]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       OPPGAVE A          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%


Ekinetisk = (VEKT*FARTB^2)/2; %Kinetisk energi [Joule]

fprintf('a) Den kinetiske energien er %g KJoule.\n\n',
 fix(Ekinetisk/1000))



%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       OPPGAVE B          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

TID = (FARTB/FARTA);    % FARTA = FARTB, løser med hensyn på T
A = (1/2)*FARTA*TID^2; % 1/2*m/s^2
B = FARTB*TID;

TOTAL = A+B;

fprintf('b) Strekningen må være %g m for at de skal få samme fart.\n',
 TOTAL)

if(A>B) fprintf('   Bil A kjører lengst, %g m.\n\n', A)
end
if(A<B) fprintf('   Bil B kjører lengst, %g m.\n\n', B)
end



%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       OPPGAVE C          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

FARTA = 2.7; %m/s^2

v = 75/3.6; %m/s

v0 = 0; %m/s
```

```
for TID = 1:0.01:1000
    B = v*TID;
    A = (1/2)*FARTA*TID^2;

    if(A>B)

        break;
    end

end

fprintf('c) Bil A passerer Bil B etter %g sekunder.\n\n\n\n',TID)
```

a) Den kinetiske energien er 282 KJoule.

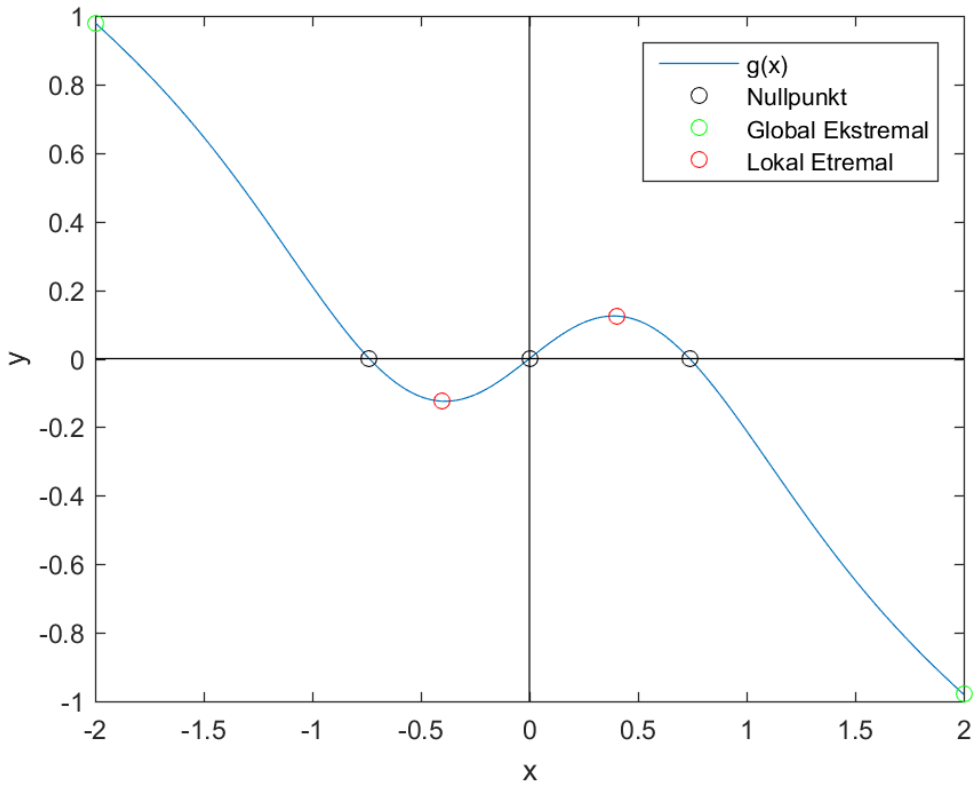
b) Strekningen må være 241.127 m for at de skal få samme fart.
Bil B kjører lengst, 160.751 m.

c) Bil A passerer Bil B etter 15.44 sekunder.

Published with MATLAB® R2015b

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% OPPGAVE 2           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


x = -2:0.001:2;
g = atan(x).*exp(-x.^2)-(x/2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

plot(x,g), hold on

for n = 1:length(g)
    if(g(n)<0)
        break
    end
end

for m = n+1:length(g)
    if(g(m)>0)
        break
    end
end

for o = m+1:length(g)
    if(g(o)<0)
        break
    end
end

plot([x(n) x(m) x(o)], [0 0 0], 'ok')

fprintf('-----------------OPPGAVE 2-----------------\n\n')
fprintf('Nullpunkter: (0,%g), (0, %g), (0, %g)\n',x(n),x(m),x(o))


x = -2:0.1:2;
g = atan(x).*exp(-x.^2)-(x/2);

g_max = g(1);
x_max = x(1);
g_min = g(1);
x_min = x(1);

for i = 1:length(g)
    if (g(i)>g_max)
        g_max = g(i);
        x_max = x(i);
    elseif (g(i)<g_min)
        g_min = g(i);
        x_min = x(i);
```

```
    end
end

plot([x_max x_min], [g_max g_min], 'og')
fprintf('Globale ekstremalpunkt: (%g, %g), (%g,%g)\n',x_max, g_max,
 x_min, g_min)


x = -0.7:0.1:0.7;
g = atan(x).*exp(-x.^2)-(x/2);

g_max = g(1);
x_max = x(1);
g_min = g(1);
x_min = x(1);

for i = 1:length(g)
    if (g(i)>g_max)
        g_max = g(i);
        x_max = x(i);
    elseif (g(i)<g_min)
        g_min = g(i);
        x_min = x(i);

    end
end

plot([x_max x_min], [g_max g_min], 'or')
fprintf('Lokale ekstremalpunkt: (%g, %g), (%g, %g)\n',x_max, g_max,
 x_min, g_min)
legend('g(x)','Nullpunkt' ,'Global Ekstremal', 'Lokal Etremal')
line(xlim, [0 0],'color','k')%x-akse
line([0 0], ylim,'color','k')%y-akse
ylabel('y') % label for x axis
xlabel('x') % label for y axis
hold off

----------------OPPGAVE 2----------------

Nullpunkter: (0,-0.737), (0, 0.001), (0, 0.738)
Globale ekstremalpunkt: (-2, 0.979722), (2,-0.979722)
Lokale ekstremalpunkt: (0.4, 0.124246), (-0.4, -0.124246)
```



Published with MATLAB® R2015b

Areal ved hjelp av simpsons metode

```
%%%%%%%%%%%%%%%%%%%%%%%%%
%  f.m funksjonsfil:    %
%   function y = f( x ) %
%     y = 1/(1+x^2);    %
%   end                 %
%%%%%%%%%%%%%%%%%%%%%%%%%
clc, clear
a    = 0;
b    = 3;
areal= 0;

k  = 5;
dx = (b-a)/(2*k) %Skivebredden dx (nøyaktighet)

for m=1:k
 %Beregner funksjonsverdien
 f1 = f(a+(2*m-2)*dx);
  f2 = f(a+(2*m-1)*dx);
   f3 = f(a+2*m*dx);
 sum = (f1+(4*(f2))+f3 )*dx/3;
 areal  = areal + sum;
end
fprintf('Arealet er: %.10f \n\n', areal);


% Beregner analytisk verdi
syms fx
fx = 1/(1+fx^2);
fx = matlabFunction(fx);
analytisk_areal = integral(fx,0,3)


feilprosent = (analytisk_areal-areal)/analytisk_areal;
feilprosent = feilprosent*100;

fprintf('Feilprosent: %.3f%% \n\n', feilprosent)


dx =

    0.3000

Arealet er: 1.2490135802


analytisk_areal =

    1.2490

Feilprosent: 0.003%
```

Published with MATLAB® R2015b

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%% OPPGAVE 4          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear a
a(1) = 1;
a(2) = 2;
A = 0.3;
B = 0.7;

for n = 3:10
    a(n) = A*a(n-1) + B*(n-2);
end
b = mat2str(a,3);
fprintf('Oppgave 4a : %s\n',b)


syms C D lambda
k = lambda^2 - C*lambda - D;
k = solve(k);
k5 = matlabFunction(k);
k1 = matlabFunction(k(1));
k2 = matlabFunction(k(2));
clear C D lambda


lambda1 = k1(A,B);
lambda2 = k2(A,B);
fprintf('Oppgave 4b: Den generelle løsningen er: A*%.3f^n+B*
%.3f^n',lambda1,lambda2)

total_matrise = [   lambda1^a(1) lambda2^a(1) 1
                    lambda1^a(2) lambda2^a(2) 2   ];
rref(total_matrise)
```

```
Oppgave 4a : [1 2 1.3 1.79 2.64 3.59 4.58 5.57 6.57 7.57]
Oppgave 4b: Den generelle løsningen er: A*-0.700^n+B*1.000^n
ans =

    1.0000         0    0.8403
         0    1.0000    1.5882
```

Published with MATLAB® R2015b