

```

% Signalbehandling for Computer-ingeniører, CT-5, 2. forelæsning

% Dette program beregner foldnings-summen af to sekvenser, x_1 og x_2.
% Foldning beregnes vha. to nestede løkker, hvor den ydre opdaterer
% sample-nummeret, mens den indre beregner selve produkt-summen.

clear;

% Vi starter med at definere de to kausale sekvenser hver med længden L.

L = 200; %Det er ikke så vigtigt, hvilken værdi L sættes til -- den skal bare
        %større end sekvens-længderne angivet i opgaven.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Spm. a
%for i=0:4
%    x_1(i+1) = 1;
%end
%for i=5:(L-1)
%    x_1(i+1) = 0;
%end
%
%for i=0:9
%    x_2(i+1) = 1;
%end
%for i=10:(L-1)
%    x_2(i+1) = 0;
%end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Spm. b
%for i=0:4
%    x_1(i+1) = i;
%end
%for i=5:(L-1)
%    x_1(i+1) = 0;
%end
%
%for i=0:9
%    x_2(i+1) = 1;
%end
%for i=10:(L-1)
%    x_2(i+1) = 0;
%end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Spm. c
%for i=0:L
%    x_1(i+1) = exp(-i/2);
%end
%
%for i=0:9
%    x_2(i+1) = 1;
%end
%for i=10:(L-1)
%    x_2(i+1) = 0;
%end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Spm. d
%for i=0:L
%    x_1(i+1) = exp(-i/2);
%end
%
%for i=0:9
%    x_2(i+1) = i;
%end
%for i=10:(L-1)

```

```

%      x_2(i+1) = 0;
%end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Spm. e
for i=0:L
    x_1(i+1) = 0.125*exp(-0.125*i);
end

for i=0:L
    x_2(i+1) = sin(0.39*i);
end

max_samples = 2*L; % Hjelpe-variabel ifm. produkt-summering. Idet begge
                   % sekvenser har længden L, kan der ikke forekomme mere
                   % end 2L samples forskellig fra nul.

for n = 0:(max_samples-1) %n er sample-nummeret og definerer den ydre løkke

    sum = 0; % sum er en variabel der benyttes til beregning af produkt-summen
             % for sample nummer n, og sættes følgelig lig nul initialt.

    % Den indre løkke beregner selve produktsummen og skal i teorien
    % udregnes i intervallet fra minus uendelig til plus uendelig, hvilket
    % imidlertid ikke er muligt. I stedet beregnes og summeres et endelig
    % antal produkter svarende til de faktiske overlap mellem de to
    % sekvenser (hvilket vi kan beregne, qua at sekvenserne er kausale).

    if n<L
        overlap = n+1;
        for k=1:overlap
            sum = sum + x_1(k) * x_2((n+1)-k+1);
        end
        conv_sum(n+1) = sum;
    end

    if n>=L
        overlap = 2*L-n-1;
        for k=1:overlap
            sum = sum + x_1(k+1) * x_2(L-k);
        end
        conv_sum(n+1) = sum;
    end

end

for i=0:2*L-1
    x_axis(i+1)=i;
end

plot(x_axis(1:200), conv_sum(1:200));

```