```
clc
clear
%problem 1.1

%a priori probabilities
Px = [ 0.4 0.2 0.2 0.1 0.05 0.05];

% a)
% self information
Ix = -log2(Px)

%entropy
Hx = sum(Px.*Ix)

% b)
% for at få max entropy, skal alle symboler have samme probability
PxMaxEntropy = [ 1/6 1/6 1/6 1/6 1/6 1/6];

% self information
IxMaxEnt = -log2(PxMaxEntropy);

%max entropy
HxMax = sum(PxMaxEntropy.*IxMaxEnt)

%efficiency
eff = (Hx/HxMax) * 100

clear
%problem 1.2
Px = [ 1/2 1/4 1/8 1/16 1/16];

% a)
% self information
Ix = -log2(Px);

%entropy
Hx = sum(Px.*Ix)

% b)
Iseq_DADED = Ix(4) + Ix(1) + Ix(4) + Ix(5) + Ix(4)

clear
%problem 1.7
Pch = [ 3/5 2/5 ; 1/5 4/5]

Px = [ 0.5 0.5 ]
% self information
Ix = -log2(Px)

%a priori entropy
Hx = sum(Px.*Ix)
```

```
%probability that a given symbol is the output
%Py0 = Pch(1,1)*Px(1) + Pch(2,1)*Px(2);
%Py1 = Pch(1,2)*Px(1) + Pch(2,2)*Px(2);
%Py = [ Py0 Py1 ]

%dette kode gør det samme som det udkommenterede ovenfor
[nInputs, nOutputs] = size(Pch);
Py = zeros(1,nOutputs);
for i = 1:nOutputs
    Py(i) = sum((Pch(:,i)').*Px);
end

%a posteriori probabilities
PaPosteriori = zeros(nInputs, nOutputs);
for i = 1:nInputs
    for j = 1:nOutputs
       PaPosteriori(j, i) = (Pch(i,j)*Px(i))/Py(j);
    end
end

% self information
Iy = -log2(PaPosteriori);

%a posteriori entropy
Hy = zeros(1, nOutputs);
temp = PaPosteriori .* Iy;
for i = 1:nOutputs
   Hy(i) = sum(temp(i,:));
end
Hy


clear
```

```
        Ix =

            1.3219    2.3219    2.3219    3.3219    4.3219    4.3219


        Hx =

            2.2219


        HxMax =

            2.5850


        eff =

           85.9559
```

```
Hx =

    1.8750


Iseq_DADED =

    17


Pch =

    0.6000    0.4000
    0.2000    0.8000


Px =

    0.5000    0.5000


Ix =

     1     1


Hx =

     1


Hy =

    0.8113    0.9183
```

Published with MATLAB® R2013b