Problem 1.

```
% OFDM receiver

clear all; close all; clc;

% System parameters
num_carriers = 256;     % total number of carriers
num_zeros = 5;          % number of unused carriers
                        %(at each end of the frequency spectrum)
prefix_length = 25;     % length of the cyclic prefix
num_ofdm_symbols = 2; % number of OFDM symbols per frame
                      %(1 will be used to transmit the preamble and the rest for dat

%load the data
load ofdm_ex

%Implement the OFDM receiver

% remove the cylic prefix:
rx_withCP = reshape(rx_signal((1 : num_ofdm_symbols *...
... (num_carriers + prefix_length))), prefix_length+num_carriers, num_ofdm_symbols);
rx_noCP = rx_withCP(prefix_length+(1:num_carriers), :);

% go to the frequency domain
Rf = fft(rx_noCP, num_carriers);

% remove the zero carriers
Rf = Rf(num_zeros+1:end-num_zeros, :);

%estimate channel coefficients (frequency domain)
preamble_rx=Rf(:,1);
lambda=preamble_rx./preamble;

%perform data equalization
data_rx=Rf(:,2:end);
data_eq=data_rx./repmat(lambda,1,num_ofdm_symbols-1);

data_eq=sign(real(data_eq));
```

Problem 2.

```
% determine the complete subframes from a sequence of received bits

clear all; close all; clc;

% Load the data received from the satellite and the preamble
load preamble_ex

%minimum number of complete frames
Nbits=numel(data);
nFull=floor(Nbits/lFrame)-1;

preambleRep=repmat([preamble, zeros(1,lFrame-length(preamble))], 1, nFull);

pCorr=round(abs(xcorr(data, preambleRep)));

%find the start of the first frame
pCorr(1:length(data)-1)=[];
ind=find(pCorr==max(pCorr));
ind=ind(1);

%remove excess bits at the beginning
data_s=data(ind:end);

%number of complete subframes
n_sf=floor(numel(data_s)/lFrame);

%remove excess bits at the end
data_s=data_s(1:n_sf*lFrame);

%flip bits if necessary
if data_s(1:8)~=preamble
    data_s=-data_s;
end

%number of bits removed at the start and at the end
bits_s=ind-1
bits_e=mod(length(data)-bits_s,lFrame)
```

Problem 3.

```
% initialize the vector containing max values
m=zeros(1, numel(Dshift));
...
%xcorr makes more computations than it is necessary
%we can use the inner product
     m(i)=abs(sum(data.*CA));
    %or m(i)=abd(data*transp(CA));
...
pos=[1, 51, 101, 151, 201];
v=m(pos);
```

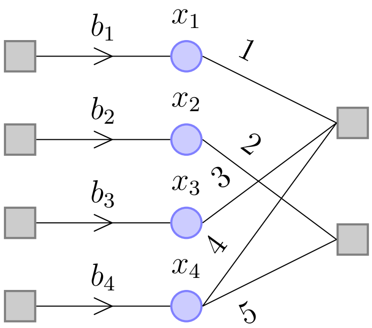
Problem 4.

$$H = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

(a)

$$Hx^T = 0$$

$$x = (1101)$$



(b)

(c)

$$Mc = \begin{pmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 1 \\ 4 & 1 \\ 4 & 2 \end{pmatrix}$$

(d)

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

(e)

$$b_i = \ln\left(\frac{P_{Y|X}(y_i|1)}{P_{Y|X}(y_i|0)}\right)$$

Problem 5.

(a)

$$\begin{pmatrix} +\cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ +\sin(\alpha) & 0 & +\cos(\alpha) \end{pmatrix}$$

(b) 6 parameters

(c) 4 satellites (we have 4 unknowns: posititionwith respect to the 3 axes, and receiver clock offset)

(d)

$$p_{13} = R_z(\dot{\Omega} \cdot 3600) \cdot p_{12}$$

(e)

(i) The distance between the satellite pposition at t_{tr} and the receiver position at t_1

(ii) $d = c(t_2 - t_{tr} - b \cdot T_b)$

(iii)

$$b \cdot T_b = (1+\nu)(t2 - t1) \Rightarrow \nu = \frac{b \cdot T_b}{t2 - t1} - 1$$

$$\text{Because } \nu = \frac{v}{c}, \Rightarrow v = \frac{c \cdot b \cdot T_b}{t2 - t1} - c$$