```
lam = x(1:(p+q),1);
alpha = x((p+q+1):2*p+q,1);
beta = x(2*p+q+1:2*(p+q),1);
w = -X*lam;
nw = sqrt(w'*w);   % norm of w
fprintf('nw =  %d \n',nw)
lamb = x(1:p,1);
mu = x(p+1:p+q,1);
b = -(sum(lamb) - sum(mu));
fprintf('b =  %.15f \n',b)
%

tols = 10^(-10); tolh = 10^(-9);
% tols < lambda_i < K - tolh
[lambnz,numsvl1] = findpsv2(lamb,K,tols,tolh);
% tols < mu_i < K - tolh
[munz,numsvm1] = findpsv2(mu,K,tols,tolh);
fprintf('numsvl1 = %d ',numsvl1)
fprintf('    numsvm1 = %d \n',numsvm1)
%  lambda_i >= K - tolh
[lamK,pf] = countumf2(lamb,K,tolh);  % number of blue margin failures
%  mu_j >= K - tolh
[muK,qf] = countvmf2(mu,K,tolh);  % number of red margin failures
fprintf('pf =  %d ',pf)
fprintf('   qf =  %d \n',qf)
[~,pm] = countmlu2(lamb,tols);  % number of  points such that lambda_i > tols
[~,qm] = countmlv2(mu,tols);   % number of  points such that mu_i > 0
fprintf('pm =  %d ',pm)
fprintf('   qm =  %d \n',qm)
fprintf('p - pm =  %d ',p - pm)
fprintf('   q - qm =  %d \n',q - qm)
lnu =  (pf + qf)/(p+q); unu = (pm + qm)/(p+q);
fprintf('lnu =  %d ',lnu)
   fprintf('    unu =  %d \n',unu)
if nu < lnu
      fprintf('** Warning; nu is too small ** \n')
else
      if nu > unu
           fprintf('** Warning; nu is too big ** \n')
      end
end
```