1. (a)
Given prior on [-m, m]:
$\pi(\Theta) = cos^2(\pi\Theta/2m)/m$
For m= 2:
$\pi(\Theta) = cos^2(\pi\Theta/4)/2$

Given normal distribution from which samples were observed:
$f(y|\Theta) \propto \sqrt{\tau} * exp(-\tau/2 * (y-\Theta)^2)$

For τ = ¼:
$f(y|\Theta) \propto 1/2 * exp(-1/8 * (y-\Theta)^2)$

$posterior \propto cos^2(\pi\Theta/4)/2 * \prod^n_{i=1} 1/2 * exp(-1/8 * (y-\Theta)^2)$

Proposal:
$q(\Theta|\Theta') = U(-2,2) = 1/(b-a) = 1/4$

$\gamma = \pi(\Theta') * q(\Theta|\Theta')/\pi(\Theta) * q(\Theta'|\Theta)$
$= (cos^2(\pi\Theta'/4)/2 * \prod^n_{i=1} 1/2 * exp(-1/8 * (y_i-\Theta')^2 * 1/4))/(cos^2(\pi\Theta/4)/2 * \prod^n_{i=1} 1/2 * exp(-1/8 * (y_i-\Theta)^2) * 1/4)$
$= (cos^2(\pi\Theta'/4)/2 * \prod^n_{i=1} 1/2 * exp(-1/8 * (y_i-\Theta')^2 * 1/4))/(cos^2(\pi\Theta/4)/2 * \prod^n_{i=1} 1/2 * exp(-1/8 * (y_i-\Theta)^2) * 1/4)$
$= (cos(\pi\Theta'/4)/cos(\pi\Theta/4))^2 * \prod^n_{i=1} exp(-1/8 * ((y_i-\Theta')^2 - (y_i-\Theta)^2))$
$= (cos(\pi\Theta'/4)/cos(\pi\Theta/4))^2 * \prod^n_{i=1} exp(-1/8 * (\Theta-\Theta') * (2 * y_i - \Theta - \Theta'))$

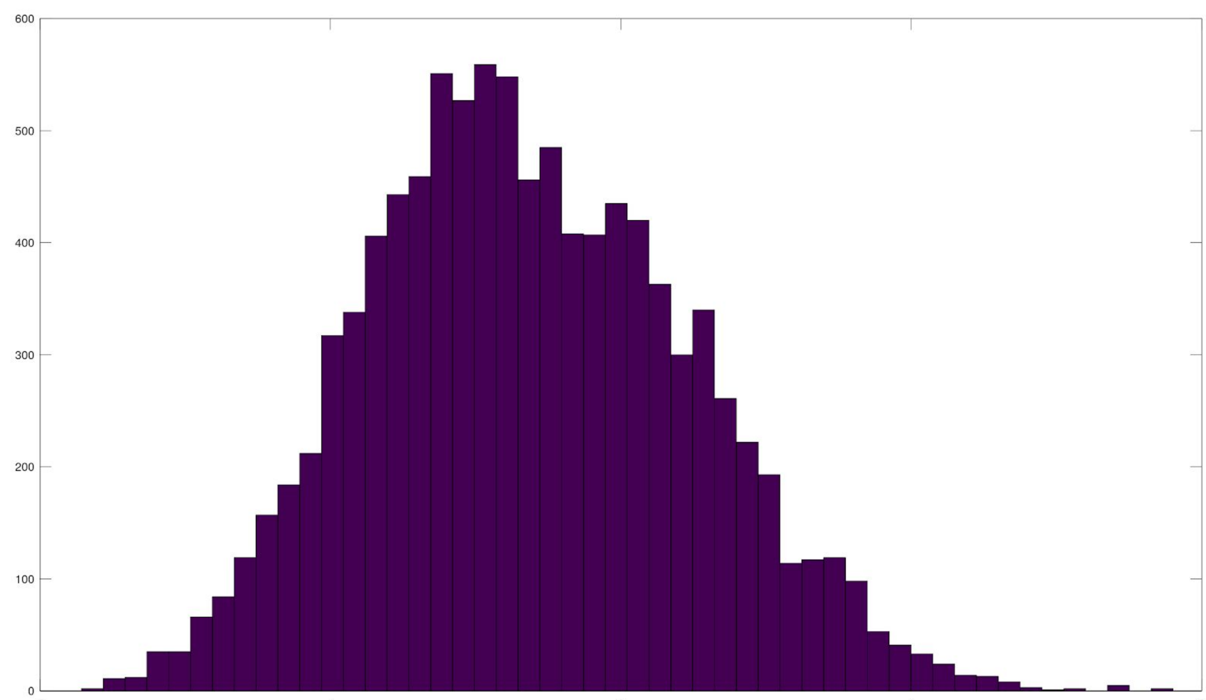
$\rho = 1 \Lambda \gamma$
$\rho = 1 \Lambda (cos(\pi\Theta'/4)/cos(\pi\Theta/4))^2 * \prod^n_{i=1} exp(-1/8 * (\Theta-\Theta') * (2 * y_i - \Theta - \Theta'))$

For my use case, I'll take the initial value of Θ as 0.

I implemented below metropolis algorithm in Octave: (The file is also attached with name q1.m) q1.m

```
close all force
clear all

data = [-2.00 -3.00 4.00 -7.00 0.00 4.00];

theta = 0;

thetas = [theta];

tic
for i = 1:10500
  theta_prop = -2 + (4 * rand)
%-------------------------------------------------------------------------
  prod1 = prod(exp((-1/8)*(theta - theta_prop)*(data.*2 - theta_prop - theta)))
%-------------------------------------------------------------------------
  r = ((cos((pi * theta_prop)/4)/cos((pi * theta)/4))^2)*prod1;
%-------------------------------------------------------------------------
rho = min(r ,1);
   if (rand < rho)
       theta = theta_prop;
   end
thetas =  [thetas theta];
end
toc
%Burn in 500
 thetas = thetas(500:end);
figure(1)
 hist(thetas, 50)
 mean(thetas)
 var(thetas)
 prctile(thetas,2.5)
 prctile(thetas,97.5)
```

(a) Plot obtained for 10,000 observation from posterior after discarding first 500 observations is:



(b) Following are the results that I got from my matlab code. They results are in order of mean, variance, 2.5 percentile, 97.5 percentile:

```
ans = -0.31967
ans =  0.30763
ans = -1.3266
ans =  0.80501
```

Thus,
Bayes estimator of Θ is: -0.31967
95% equitalied credible set is: [-1.3266, 0.80501]

2.
High protein diet:
$f(y_1|\Theta_1, \tau_1) \propto \sqrt{\tau_1} * exp(-\tau_1/2 * (y_{1i} - \Theta_1)^2)$
This means that:
$Y_{11}, Y_{12}, ..., Y_{1n} \sim N(\Theta_1, 1/\tau_1)$

$\Theta_1 \sim N(\Theta_{10}, 1/\tau_{10})$
$\tau_1 \sim Ga(a_1, b_1)$

The joint distribution is given by:
$f(y_1|\Theta_1, \tau_1) = \prod^n_{i=1} f(y_{1i}|\Theta_1, \tau_1) * \pi(\Theta_1) * \pi(\tau_1)$
$\propto \tau_1^{n/2} exp(\tau_1/2 * \sum_{i=1}^n (y_{1i} - \Theta_1)^2) * exp(-(\Theta_1 - \Theta_{10})^2/2) * \tau_1^{a_1-1} exp(-b_1\tau_1)$

To find the full conditional probability of Θ_1, we select terms from $f(y_1|\Theta_1, \tau_1)$ that contains Θ_1 and normalize
$\pi(\Theta_1 | \tau_1, y_1) = \pi(\Theta_1, \tau_1|y_1) / \pi(\tau_1|y_1) = \pi(\Theta_1, \tau_1, y_1) / \pi(\tau_1, y_1) \propto \pi(\Theta_1, \tau_1, y_1)$

Thus,
$\pi(\Theta_1 | \tau_1, y_1) \propto exp\{-\tau_1/2 * \sum_{i=1}^n (y_{1i} - \Theta_1)^2\} * exp\{\tau_{10}/2 * (\Theta_1 - \Theta_{10})^2\}$
$\propto exp\{-1/2 * (\tau_{10} + n_1\tau_1) * (\Theta_1 - (\tau_1 \sum_{i=1}^n y_{1i} + \Theta_{10}\tau_{10})/(\tau_{10} + n_1\tau_1))^2\}$

From the above expression, we can see that this is a kernel of :
$N(\{\tau_1 \sum_{i=1}^n y_{1i} + \Theta_{10}\tau_{10}\}/\{\tau_{10} + n_1\tau_1\}, 1/(\tau_{10} + n_1\tau_1))$
Where:
y_{1i} = [134, 146, 104, 119, 124, 161, 107, 83, 113, 129, 97, 123]
$\Theta_{10} = 110$
$\tau_{10} = 1/100$
$n_1 = 12$

(I am not putting the exact values of these constants in the kernel, as this is not asked in the question. My Octave code will calculate the exact values.)
Similarly,
$\pi(\tau_1 | \Theta_1, y_1) \propto \tau_1^{n/2} * exp\{-\tau_1/2 * \sum_{i=1}^{n}(y_{1i} - \Theta_1)^2\}\tau_1^{a_1-1} * exp(-b_1\tau_1)$
$= \tau_1^{n/2+a_1-1} * exp\{-\tau_1[b_1 + 1/2 * \sum_{i=1}^{n}(y_{1i} - \Theta_1)^2]\}$
Clearly the above expression is the kernel of:
$Ga(a_1 + n_1/2,\ b_1 + 1/2 * \sum_{i=1}^{n}(y_{1i} - \Theta_1)^2)$
Where:
y_{1i} = [134, 146, 104, 119, 124, 161, 107, 83, 113, 129, 97, 123]
$a_1 = 0.01$
$b_1 = 4$
$n_1 = 12$
(I am not putting the exact values of these constants in the kernel, as this is not asked in the question. My Octave code will calculate the exact values.)

Similarly, for low protein diet, the kernels are given as follows:
(I won't need to derive this again, as they consist of the same family of distributions as high protein diet distributions mentioned above.)
$\pi(\Theta_2 | \tau_2, y_2)$ has a kernel of: $N(\{\tau_2\sum_{i=1}^{n}y_{2i} + \Theta_{20}\tau_{20}\}/\{\tau_{20} + n_2\tau_2\},\ 1/(\tau_{20} + n_2\tau_2))$
y_{2i} = [70, 118, 101, 85, 107, 132, 94]
$\Theta_{20} = 110$
$\tau_{20} = 1/100$
$n_2 = 7$
(I am not putting the exact values of these constants in the kernel, as this is not asked in the question. My Octave code will calculate the exact values.)

$\pi(\tau_2 | \Theta_2, y_2)$ has a kernel of: $Ga(a_2 + n_2/2,\ b_2 + 1/2 * \sum_{i=1}^{n}(y_{2i} - \Theta_2)^2)$
Where:
y_{2i} = [70, 118, 101, 85, 107, 132, 94]
$a_2 = 0.01$
$b_2 = 4$
$n_2 = 7$
(I am not putting the exact values of these constants in the kernel, as this is not asked in the question. My Octave code will calculate the exact values.)

For sampling, I will use the original algorithm mentioned in the lectures which is given below:

$$
\begin{aligned}
&\text{Draw } \theta_1^{n+1} \text{ from } \pi(\theta_1 | \theta_2^n, \theta_3^n, \ldots, \theta_p^n)\\
&\text{Draw } \theta_2^{n+1} \text{ from } \pi(\theta_2 | \theta_1^{n+1}, \theta_3^n, \ldots, \theta_p^n)\\
&\text{Draw } \theta_3^{n+1} \text{ from } \pi(\theta_3 | \theta_1^{n+1}, \theta_2^{n+1}, \theta_4^n, \ldots, \theta_p^n)\\
&\ldots\\
&\text{Draw } \theta_{p-1}^{n+1} \text{ from } \pi(\theta_{p-1} | \theta_1^{n+1}, \theta_2^{n+1}, \ldots, \theta_{p-2}^{n+1}, \theta_p^n)\\
&\text{Draw } \theta_p^{n+1} \text{ from } \pi(\theta_p | \theta_1^{n+1}, \theta_2^{n+1}, \ldots, \theta_{p-1}^{n+1})
\end{aligned}
$$

For my case, I'll sample Θ_1 and τ_1 independently from one another and fixing one of them at a time. And, similarly Θ_2 and τ_2 independently from one another and also independently from Θ_1 and τ_1 as they are not related and are in fact independent from one another. This is because low protein diet and high protein diet were given independently from each other.

Below is my exact code which I used for sampling using Octave (also attached with filename q2.m if you would like to run it). Note that I have set my initial values of $\Theta_1 = 110$ and $\tau_1 = 1/100$; $\Theta_2 = 110$ and $\tau_2 = 1/100$ (The initial values won't matter much because we are going to burn 500 samples anyway):

```
pkg load statistics
close all
clear all
 randn("state",1)
 randg("state",1)
 y1 = [134.00 146.00 104.00 119.00 124.00 161.00 107.00 83.00 113.00 129.00 97.00 123.00];
 n1 = length(y1);
 y2 = [70.00 118.00 101.00 85.00 107.00 132.00 94.00]
 n2 = length(y2);
%-------------------------------------------
NN = 10000;
thetas1 = []; taus1 = [];
thetas2 = []; taus2 = [];
sumdata1 = sum(y1);
sumdata2 = sum(y2);
%hyperparameters
theta10= 110; tau10 = 1/100;
theta20= 110; tau20 = 1/100;
a1= 0.01; b1= 4;
a2= 0.01; b2= 4;
% start, initial values
theta1 = 110;   tau1 = 1/100; %
theta2 = 110;   tau2 = 1/100; %
for i = 1 : NN
  newtheta1  = sqrt(1/(tau10+1*tau1)) * randn + (tau1 * sumdata1+tau10*theta10)/(tau10+n1*tau1);
  %par   = b+1/2 * sum ( (y1 - theta).^2);
  par1   = b1+1/2 * sum ( (y1 - newtheta1).^2);
  newtau1 = gamrnd(a1 + n1/2, 1/par1); %par is rate
  thetas1 = [thetas1 newtheta1];
  taus1 = [taus1 newtau1];
  theta1=newtheta1;
  tau1=newtau1;

  newtheta2  = sqrt(1/(tau20+1*tau2)) * randn + (tau2 * sumdata2+tau20*theta20)/(tau20+n2*tau2);
  %par   = b+1/2 * sum ( (y2 - theta).^2);
  par2   = b2+1/2 * sum ( (y2 - newtheta2).^2);
  newtau2 = gamrnd(a2 + n2/2, 1/par2); %par is rate
  thetas2 = [thetas2 newtheta2];
  taus2 = [taus2 newtau2];
  theta2=newtheta2;
  tau2=newtau2;

end

burn =500;
thetas1 = thetas1(burn+1:end);
taus1=taus1(burn+1:end);

thetas2 = thetas2(burn+1:end);
taus2=taus2(burn+1:end);

thetas1minusthetas2 = thetas1 - thetas2;
mean(thetas1minusthetas2)
length(thetas1minusthetas2(thetas1minusthetas2>0))
 prctile(thetas1minusthetas2,2.5)
 prctile(thetas1minusthetas2,97.5)
mean(thetas1)
mean(taus1)
mean(thetas2)
mean(taus2)
```

Note above that although Θ_1 and Θ_2 are inside the same 'for' loop, the logic for sampling them is entirely independent of one another. They do not share any variable.

To explain more, in the above logic, I have followed the below steps for High protein diet case:

a) I find full conditionals independently for high protein diet.
b) I form a kernel of joint distribution of all parameters and data.
c) I find the full conditional for all components. Θ_1 has kernel $N(\{\tau_1 \sum_{i=1}^{n} y_{1i} + \Theta_{10}\tau_{10}\}/\{\tau_{10} + n_1\tau_1\},\ 1/(\tau_{10} + n_1\tau_1))$ where other variables such as τ_1 are kept constant. Similarly, τ_1 has kernel $Ga(a_1 + n_1/2,\ b_1 + 1/2 * \sum_{i=1}^{n}(y_{1i} - \Theta_1)^2)$ where other variable such as Θ_1 is kept constant

d) Normalize each component as a distribution.
e) Sample from both conditionals Θ_1 and τ_1 .

We do the same steps for low protein diet case.

The above code gave me following result after sampling:

```
ans =  116.59
ans =  0.0020191
ans =  104.92
ans =  0.0022806
```

Here, bayes estimate of Θ_1 is 116.59
Here, bayes estimate of τ_1 is 0.0020191
Here, bayes estimate of Θ_2 is 104.92
Here, bayes estimate of τ_2 is 0.0022806

(b) The below output I got from my sampling code. Their results are in the order of:
Mean of ($\Theta_1 - \Theta_2$), number of samples out of 9500 which have $\Theta_1 - \Theta_2 > 0$, 2.5 percentile of $\Theta_1 - \Theta_2$, 97.5 percentile of $\Theta_1 - \Theta_2$, mean of Θ_1, mean of τ_1, mean of Θ_2, mean of τ_2

```
ans =  11.665
ans =  7739
ans = -13.833
ans =  37.448
ans =  116.59
ans =  0.0020191
ans =  104.92
ans =  0.0022806
```

Therefore,
i) The sample difference: $\Theta_1 - \Theta_2$ is given by: 11.665
ii) Proportion of positive differences approximates the posterior probability of hypothesis H_0 $\Theta_1 > \Theta_2$ is given by: 7739/9500 = 0.81463157894 ~ 81.46%

(c) From my Octave code, I found the below values for 2.5 percentile and 97.5 percentile for my equitailed credible set:

```
ans = -13.833
ans =  37.448
```

Therefore 95% equitailed credible set for $\Theta_1 - \Theta_2$ is given by [-13.833, 37.448].
Yes, my 95% equitailed credible set contains 0.