1.1 Beta

Implement a dynamic programming function calculating coefficients beta for a hidden Markov model. See Slide 27 of Class 2 for a definition.

Use the following header (or a close equivalent if you are using R).

```
function b = beta_dynamic(M,p,B,v)

% BETA_DYNAMIC(M,p,B,v) calculates the matrix of betas for the hmm with
% transition matrix M, emission matrix B, and initial probabilities
% p, given the observations v
```

This function will output the $N \times T$ matrix of betas.

[10 marks]

1.2 Gamma

Implement a function calculating coefficients gamma for a hidden Markov model. See Slide 28 of Class 2 for a definition.

Use the following header (or a close equivalent if you are using R).

```
function g = gamma_dynamic(alpha,beta)

% GAMMA_DYNAMIC(alpha,beta) calculate gamma for hmm given alpha and beta
```

The header is written in this way so that one could run the function as follows: `g = gamma_dynamic(alpha_dynamic(M,p,B,v),beta_dynamic(M,p,B,v))`.

This function will output the $N \times T$ matrix of gammas.

[10 marks]

1.3 HMM Example

The files `M.txt`, `B.txt`, and `p.txt` contain the transition matrix, the emission matrix, and the initial probabilities for a hmm with 12 hidden states and 10 observable states. The file `v.txt` contains a sequence of 20 observed states. You can read the files using `M = dlmread('M.txt')` etc.

Run your functions on this data and calculate the matrices of betas and gammas. Extract the values $\beta_{18}(3)$ and $\gamma_{10}(5)$.

[4 marks]