Exercise 1:

Code:

```
A = [0, 1/3, 1/2, 1/2, 0; 1/2, 0, 0, 0, 0; 0, 1/3, 0, 0, 0; 0, 1/3, 1/2, 0, 1; 1/2,
0, 0, 1/2, 0];

p0 = [0;1;0;0;0];

result = (A^100)*p0;
disp("When A^100:")
disp(result)

result = (A^200)*p0;
disp("When A^200:")
disp(result)

result = (A^500)*p0;
disp("When A^500:")
disp(result)
```

Result:

When A^100:

0.2264
0.1132
0.0377
0.3396
0.2830

When A^200:

0.2264
0.1132
0.0377
0.3396
0.2830

When A^500:

0.2264
0.1132
0.0377
0.3396
0.2830

Explanation:

This code computes and shows the probabilities after 100, 200, and 500 steps starting from website 2. There is no change in the probability regardless of how many times it is visited.

Exercise 2:

Code:

```
A = [0, 1/3, 1/2, 1/2, 0; 1/2, 0, 0, 0, 0; 0, 1/3, 0, 0, 0; 0, 1/3, 1/2, 0, 1; 1/2,
0, 0, 1/2, 0];

p0 = [0;0;0;1;0];

result = (A^100)*p0;
disp("When A^100:")
disp(result)

result = (A^200)*p0;
disp("When A^200:")
disp(result)

result = (A^500)*p0;
disp("When A^500:")
disp(result)
```

Result:

When A^100:

0.2264
0.1132
0.0377
0.3396
0.2830

When A^200:

0.2264
0.1132
0.0377
0.3396
0.2830

When A^500:

0.2264
0.1132
0.0377
0.3396
0.2830

Explanation:

This code computes and shows the probabilities after 100, 200, and 500 steps starting from website 3. There is no change in the probability regardless of how many times it is visited and is identical to the probability in exercise 1, hence showing conversion to a single vector.

Problem:

Code:

```
% Forming link matrix

A = [0, 1/2, 0, 0, 0, 0;
    0, 0, 1/3, 0, 1/2, 0;
    1/4, 1/2, 0, 0, 0, 0;
    1/4, 0, 0, 0, 1/2, 0;
    1/4, 0, 1/3, 0, 0, 1;
    1/4, 0, 1/3, 1, 0, 0];

% Calcultaion of power by raising A to 100 and multiplying with p0
p0 = [0; 0; 1; 0; 0; 0];
power_result = (A^100)*p0;
disp("Result for part b:")
disp(power_result);

% Calcultaion of power by finding eigen vector p
I=eye(6); R=rref(A-I);
v=-R(:,6);v(6)=1;
p=v/sum(v);
disp("Result for part c:")
disp(p);
```

Results:

Result for part b:

0.0851
0.1702
0.1064
0.1560
0.2695
0.2128

Result for part c:

0.0851
0.1702
0.1064
0.1560
0.2695
0.2128

Explanation:

This code calculates the long-term probabilities of transitioning between states in a Markov chain represented by matrix A. It uses two methods: raising A to the power of 100 and multiplying with an initial state vector and finding the eigenvector corresponding to the eigenvalue 1. Both methods yield the same long-term probabilities.