```
% Problem 8
% 8.(a)
A = [0 0 0 0 1 0;
0 0 0 1 0 0;
0 1 0 0 0 0;
1 0 0 0 1 0;
0 1 0 0 0 1;
0 0 1 0 1 0]

A =

     0     0     0     0     1     0
     0     0     0     1     0     0
     0     1     0     0     0     0
     1     0     0     0     1     0
     0     1     0     0     0     1
     0     0     1     0     1     0

% 8.(b)
% path represented by matrix
A ^ 2

ans =

     0     1     0     0     0     1
     1     0     0     0     1     0
     0     0     0     1     0     0
     0     1     0     0     1     1
     0     0     1     1     1     0
     0     2     0     0     0     1
% specified path routes
% function used in getting 8.(b) specified paths
function L2 = deep2(A, n, l, t)
    if (l == 2)
        L2 = [t n];
    else
        temp_ans = [];
        for i = 1:6
            if (A(n,i) == 1)
                temp_ans = [temp_ans; deep2(A, i, l + 1, [t n])];
            end
        end
        L2 = temp_ans;
    end
end
% script used in getting 8.(b) specified paths
path2 =

     []

for i = 1:6
path2 = [path2; deep2(A, i, 0, [])];
end
path2

path2 =

     1     5     2
     1     5     6
     2     4     1
     2     4     5
     3     2     4
```

```
     4     1     5
     4     5     2
     4     5     6
     5     2     4
     5     6     3
     5     6     5
     6     3     2
     6     5     2
     6     5     6

% 8.(c)
% path represented by matrix
A ^ 3

ans =

     0     0     1     1     1     0
     0     1     0     0     1     1
     1     0     0     0     1     0
     0     1     1     1     1     1
     1     2     0     0     1     1
     0     0     1     2     1     0
% specified path routes
% function used in getting 8.(c) specified paths
function L3 = deep3(A, n, l, t)
    if (l == 3)
        L3 = [t n];
    else
        temp_ans = [];
        for i = 1:6
            if (A(n,i) == 1)
                temp_ans = [temp_ans; deep3(A, i, l + 1, [t n])];
            end
        end
        L3 = temp_ans;
    end
end
% script used in getting 8.(c) specified paths
path3 = []

path3 =

     []

for i = 1:6
path3 = [path3; deep3(A,i,0,[])];
end
path3

path3 =

     1     5     2     4
     1     5     6     3
     1     5     6     5
     2     4     1     5
     2     4     5     2
     2     4     5     6
     3     2     4     1
     3     2     4     5
     4     1     5     2
     4     1     5     6
     4     5     2     4
```

```
     4     5     6     3
     4     5     6     5
     5     2     4     1
     5     2     4     5
     5     6     3     2
     5     6     5     2
     5     6     5     6
     6     3     2     4
     6     5     2     4
     6     5     6     3
     6     5     6     5

% 8.(d)
% path represented by matrix
A + A ^ 2 + A ^ 3

ans =

     0     1     1     1     2     1
     1     1     0     1     2     1
     1     1     0     1     1     0
     1     2     1     1     3     2
     1     3     1     1     2     2
     0     2     2     2     2     1
% specified path routes
% function used in getting 8.(d) specified paths
function L123 = deep123(A, n, l, t)
    temp_ans = [];
    if (l <= 3  && l >= 1)
        temp_ans = [t n zeros(1,3 - l)];
    end
    if (l < 3)
        for i = 1:6
            if (A(n,i) == 1)
                temp_ans = [temp_ans; deep123(A, i, l + 1, [t n])];
            end
        end
    end
    L123 = temp_ans;
end
% script used in getting 8.(d) specified paths
path123 = []

path123 =

     []

for i = 1:6
path123 = [path123;deep123(A,i,0,[])];
end
path123

path123 =

     1     5     0     0
     1     5     2     0
     1     5     2     4
     1     5     6     0
     1     5     6     3
     1     5     6     5
     2     4     0     0
     2     4     1     0
```

```
     2     4     1     5
     2     4     5     0
     2     4     5     2
     2     4     5     6
     3     2     0     0
     3     2     4     0
     3     2     4     1
     3     2     4     5
     4     1     0     0
     4     1     5     0
     4     1     5     2
     4     1     5     6
     4     5     0     0
     4     5     2     0
     4     5     2     4
     4     5     6     0
     4     5     6     3
     4     5     6     5
     5     2     0     0
     5     2     4     0
     5     2     4     1
     5     2     4     5
     5     6     0     0
     5     6     3     0
     5     6     3     2
     5     6     5     0
     5     6     5     2
     5     6     5     6
     6     3     0     0
     6     3     2     0
     6     3     2     4
     6     5     0     0
     6     5     2     0
     6     5     2     4
     6     5     6     0
     6     5     6     3
     6     5     6     5
```

```
% problem 9
% 9.(a)
A = [1 1; 0 1]

A =

     1     1
     0     1

inv(A)

ans =

     1    -1
     0     1

A2 = [1 1 0; 0 1 1; 0 0 1]

A2 =

     1     1     0
     0     1     1
     0     0     1
```

```
inv(A2)

ans =

     1    -1     1
     0     1    -1
     0     0     1

A3 = [1 1 0 0 ; 0 1 1 0; 0 0 1 1; 0 0 0 1]

A3 =

     1     1     0     0
     0     1     1     0
     0     0     1     1
     0     0     0     1

inv(A3)

ans =

     1    -1     1    -1
     0     1    -1     1
     0     0     1    -1
     0     0     0     1

A4 = [1 1 0 0 0;0 1 1 0 0;0 0 1 1 0;0 0 0 1 1;0 0 0 0 1]

A4 =

     1     1     0     0     0
     0     1     1     0     0
     0     0     1     1     0
     0     0     0     1     1
     0     0     0     0     1

inv(A4)

ans =

     1    -1     1    -1     1
     0     1    -1     1    -1
     0     0     1    -1     1
     0     0     0     1    -1
     0     0     0     0     1

% 9.(b)
% The special form for the inverses is that, there're all 0 under the main diagonal,
full of 1 on the main diagonal, full of -1 on the diagonal immediately above the main
diagonal, then full of 1 on the diagonal above the previous diagonal... The rest
diagonals both follow this iteration law (1, then -1, then 1, then -1, then 1....)
until it reaches the up-right corner,

% problem 10
% 10.(a)
A = [1 1 0 8 -1;
1 4 1 0 8;
0 4 1 0 0;
8 1 1 4 1;
-1 8 0 1 4]

A =
```

```
     1     1     0     8    -1
     1     4     1     0     8
     0     4     1     0     0
     8     1     1     4     1
    -1     8     0     1     4

b = [0;-1;0;1;0]

b =

     0
    -1
     0
     1
     0

A \ b

ans =

    0.2152
    0.1103
   -0.4412
   -0.0597
   -0.1519

% 10.(b)
cond(A)

ans =

   13.6530

% 10.(c)
rank(A)

ans =

     5

% 10.(d)
[L,U,P] = lu(A)

L =

    1.0000         0         0         0         0
   -0.1250    1.0000         0         0         0
         0    0.4923    1.0000         0         0
    0.1250    0.1077   -0.1475    1.0000         0
    0.1250    0.4769    0.8689   -0.0794    1.0000


U =

    8.0000    1.0000    1.0000    4.0000    1.0000
         0    8.1250    0.1250    1.5000    4.1250
         0         0    0.9385   -0.7385   -2.0308
         0         0         0    7.2295   -1.8689
         0         0         0         0    7.5238
```

```
P =

     0     0     0     1     0
     0     0     0     0     1
     0     0     1     0     0
     1     0     0     0     0
     0     1     0     0     0

% 10.(e)
% Yes, it permuted some rows since P is not equivalent to I_5. It means matlab has
interchanged some rows in the procedure when performing an LU factorization of the
coefficient matrix.
```