Determinants are computed with the `det` function.

```
>> A = magic(3)
>> det(A)         % returns the determinant of A
```

The classical adjoint matrix is computed with the `adjoint` function

```
>> A = magic(3)
>> adjoint(A)     % returns the adjoint matrix
```

There isn't a function for Cramer's rule in MATLAB. But we can make our own! Under the "New" dropdown in the Home tab, create a new function. This will open a new file in your workspace. In it, replace the existing code with the following function.

```
%  cramer(A,b) performs Cramer's rule on Ax=b
%  A is an nxn mnatrix, and b is a column vector of size n
%  The function outputs the solution x
function [output] = cramer(A,b)
n = length(b);
result = zeros(n,1);     % create a column vector where the result will be
stored
D = det(A);

% bail if the matrix is singular
if D==0
    error('No Solution')
end

% loop that computes det(A_k) / det(A)
for k = 1:n
    B = A;                  % store A in a temporary variable B
    B(:,k) = b;             % replace the kth column of B with b
    result(k) = det(B)/D;   % store det(A_k)/det(A) in kth entry of result
end

output = result;
```

Save the file under the Editor tab and name it "cramer.m". Now in the console you can call the function

```
>> A = magic(4)
>> b = [1 2 3 4]'
>> cramer(A,b)
```