```
close all;
clear all;
clc
```

Problem 2 Part1

```
syms a b c [3 1] % genererate 3 symbolic "vectors"

ax = joshCross(a); % call a function that creates the "cross" matrix from a
 vector
bx = joshCross(b);
cx = joshCross(c);

LHS = ax*bx*c; % set LHS and RHS to the equation I want to prove
RHS = (c.'*a)*b-(b.'*a)*c;

LHS = expand(LHS); % distrubute terms so that isequal will work. This wont
 change the logic of the expressions
RHS = expand(RHS);

disp("By using symbolic math toolbox, I was able to show that right hand side
 = left hand side where, LHS = ax*bx*c; RHS = (c.'*a)*b-(b.'*a)*c. Below is
 the displayed result of the isequal function, which will return one if the
 two arguments are the same and 0 if they are different, called with the
 parameters of RHS and LHS.")
disp("RHS == LHS?")
disp(isequal(RHS, LHS)) % check if the expressions are the same

clear LHS RHS
```

```
By using symbolic math toolbox, I was able to show that right hand side =
 left hand side where, LHS = ax*bx*c; RHS = (c.'*a)*b-(b.'*a)*c. Below is
 the displayed result of the isequal function, which will return one if the
 two arguments are the same and 0 if they are different, called with the
 parameters of RHS and LHS.
RHS == LHS?
   1
```

Part2

```
LHS = a.'*bx*c; % set LHS and RHS to the equation I want to prove
RHS = b.'*cx*a;

LHS = expand(LHS); % distrubute terms so that isequal will work. This wont
 change the logic of the expressions
RHS = expand(RHS);

disp("This problem was solved in the same way as the prior but LHS = a.'*bx*c;
 RHS = b.'*cx*a.")
disp("RHS == LHS?")
disp(isequal(RHS, LHS)) % check if the expressions are the same
```

```
This problem was solved in the same way as the prior but LHS = a.'*bx*c; RHS =
 b.'*cx*a.
RHS == LHS?
   1
```

Published with MATLAB® R2022a