```
clc;clear;close all
syms theta3 theta4 theta2

% link input
l1=300;
l2=20;
l3=235;
l4=193.2;

% input for values
theta2_in=[0:5:360];

n=size(theta2_in, 2);
```

Grashof's criteria

```
P = [l2, l3, l4, l1];
PQ = P;
[L, locL] = max(PQ);
PQ(locL) = [];
[S, locS] = min(PQ);
PQ(locS) = [];


if locS==4 && sum(PQ)>(L+S)
    TITLE = 'This is a Double-Crank Mechanism';
elseif (locS==1||locS==3) && sum(PQ)>(L+S)
    TITLE = 'This is a Rocker-Crank Mechanism';
elseif locS==2 && sum(PQ)>(L+S)
    TITLE = 'This is a Double-Rocker Mechanism';
elseif sum(PQ)==(L+S)
    TITLE = 'This is a Change Point Mechanism';
elseif sum(PQ)<(L+S)
    TITLE = 'This is a Double-Rocker Mechanism';
end
disp(TITLE);
```

```
This is a Rocker-Crank Mechanism
```

Solving for theta3 and theta4

```
eqn1 = l1 == l2*cosd(theta2) + l3*cosd(theta3) + l4*sind(theta4-90);
eqn2 = l4*cosd(theta4-90) == l2*sind(theta2) + l3*sind(theta3);

solved = solve([eqn1, eqn2], [theta3, theta4], "Real", true);
sol_t3 = solved.theta3;
sol_t4 = solved.theta4;
```

Substituting theta2 and storing values

```
ans_t3=zeros(n, 1);
```

```
ans_t4=zeros(n, 1);
ans_phi=zeros(n, 1);

idx=1;
for t2 = theta2_in
    if t2==180
        % gives NaN for precomputed expression
        eqn1=subs(eqn1, theta2, 180);
        eqn2=subs(eqn2, theta2, 180);
        temp_sol = solve([eqn1,eqn2]);
        t3=temp_sol.theta3;
        t4=temp_sol.theta4;
    else
        t3=(subs(sol_t3, theta2, t2));
        t4=(subs(sol_t4, theta2, t2));
    end


    t3=double(t3);
    t4=double(t4);
    phi=360-(t3+t4);

    % because we have two sets of solutions
    if t4(1)<=180 && t4(1)>0 && imag(t3(1))==0 && imag(t4(1))==0
        ans_t3(idx,:)=t3(1);
        ans_t4(idx,:)=t4(1);
        ans_phi(idx, :)=phi(1);
    elseif t4(2)<=180 && t4(2)>0 && imag(t3(2))==0 && imag(t4(2))==0
        ans_t3(idx,:)=t3(2);
        ans_t4(idx,:)=t4(2);
        ans_phi(idx, :)=phi(2);
    else
        % invalid solution, can't reach
        ans_t3(idx, :) = -399;
        ans_t4(idx, :) = -399;
        ans_phi(idx, :) = -399;
    end
    idx=idx+1;
end
```

Create a table to store the data

```
t2=transpose(theta2_in);
T=table(t2, ans_t3, ans_t4, ans_phi);
writetable(T,'FourBarAngles.csv','Delimiter',',','QuoteStrings',true);
```