```
%clears all variables to ensure accuracy between runs.
clearvars

%---------------------------------------------------------------------------%
%                         Predator Prey Model 1.e                           %
%                            Dakota Ewing                                   %
%---------------------------------------------------------------------------%


%                              Variables                                    %
%---------------------------------------------------------------------------%

% To is the start month.
To = 1;
% Tf is the end month.
Tf = 12;
% dt is the time step in months.
dt = .0001;
% steps is the total number of timesteps.
Steps = (Tf - To + 1)/dt;
% T is the array in which we hold all the time values. It starts at To,
% increments by the time step, and ends at the target year, Tf.
T = To:dt:Tf;

% Yo is the starting population of Tuna.
Yo = 10000;
% Po is the starting population of Sharks.
Po = 10000;
% Ho is the population of humans, which remains constant.
Ho = 500;

% Y is the array in which we hold the number of Tuna at a given time.
% We initialize it to all zeros, except for the first month
% being simulated, which is initialized to Yo.
Y(To:Steps) = 0;
Y(To) = Yo;

% P is the array in which we hold the number of Sharks at a given time.
% We initialize it to all zeros, except for the first month
% being simulated, which is initialized to Po.
P(To:Steps) = 0;
P(To) = Po;

% H is the array in which we hold the number of Humans at a given time.
% We initialize every value to Ho, because the number of humans remains
% constant throughout the simulation.
% NOTE: This array is created to simplify the process of expanding the
% simulation to run with non-constant human populations.
```

```
H(To:Steps) = Ho;

%Ky is the birth rate fraction of Tuna.
Ky = 2;
%Kh is the birth rate fraction of Humans.
Kh = 0;
%Kp is the death rate fraction of Sharks.
Kp = 0.5;

%Kyp is the death proportionality constant of the Tuna.
Kyp = 0.0001;
%Kpy is the birth proportionality constant of the Sharks.
Kpy = 0.0001;
%Khf is the fishing rate of the Humans towards both Sharks and Tuna.
Khf = 0.00025;

%Seasonal fishing variables:

% f is a baseline constant that is added to the fishing constant function.
f = 100;
% a is the amplitude of the fishing constant function.
a = 100;
% period is the period of the fishing constant function in months.
period = 4;
% p is the period in radians.
p = (7 * pi)/6;
Fs = 6;
Fe = 12;
phase = .5;
%                              Simulation                                  %
%--------------------------------------------------------------------------%

t = To;
while t <= Steps
%     % Sets the fishing constant to zero.
%     Khf = 0;
%     % Calculates the fishing constant using a cosine function, if it is
%     % currently in the fishing season.
%     %NOTE: NOT WORKING.
%     if  (Fs <= (t * dt)) && ((t * dt) <= Fe)
%         Khf = f + a * sin(p * (t * dt) + pi/phase);
%     end

    %We calculate the change in Y, P and H for a dt by subtracting the
    % number of births from the number of deaths for the species.

    %The number of births of Y is proportional to the population of Y at t.
    Ybirths = Ky * Y(t);
    %The number of deaths of Y is proportional to the number of
```

```
    % interactions between Y and P at t, plus a proportion of the
    % interactions between H and Y at t.
    Ydeaths = Kyp * Y(t) * P(t) + Khf * Y(t) * H(t);
    dy = Ybirths - Ydeaths;

    %The number of births of P is proportional to the number of
    % interactions between Y and P at t.
    Pbirths = Kpy * P(t) * Y(t);
    %The number of deaths of P is proportional to the population of P at t,
    %plus a proportion of interactions between P and H at t.
    Pdeaths = Kp * P(t) + Khf * P(t) * H(t);
    dp = Pbirths - Pdeaths;

    %The population of H does not change.
    dh = 0;

    %The populations at t + 1 are equal to the populations at t, plus the
    %change in the given population times the time step.
    Y(t + 1) = Y(t) + dy * dt;
    P(t + 1) = P(t) + dp * dt;
    H(t + 1) = H(t) + dp * dt;

    %t is incremented by 1.
    t = t + 1;
end


%                                Plots                                    %
%-------------------------------------------------------------------------%

%plots the Y and P species over time.
plot(1:Steps, P(1:Steps),1:Steps, Y(1:Steps));

%plots the Y and P species in relation to each other.
plot(Y(1:Steps), P(1:Steps));
```