Algorithm for Problem 2

1. First, modify the budworm.m code so that it solves the HIV model for a particular set of parameters.
    a. NOTE: T* has units of cell/mm^3 and VI/VNI have units of virus particles per ml; you'll need to make an adjustment in the equations to ensure that the dimensions match.
    b. For a system of three equations, the vector u (from the budworm code) is now a matrix whose first column will correspond to T* (which is now u(1)), second column will correspond to VI (which is now u(2))and third column will correspond to VNI (which is now u(3)).
    c. Input the right-hand side of each differential equation separated by a semicolon into the equation for dudt in the budworm code.
    d. Be sure to change the function name to match the file name.
    e. Make sure your code will plot the numerical solution and the given data on the same set of axes.

2. Now modify this code (under another name) so that it computes the best fit for delta to the given data.
    a. In the 'solution' function, input the data and the initial conditions for V_I and V_{NI}; be sure to use values given in the newly uploaded data file. Note that T_0 and T*(0) are not the same; in fact, T*(0) depends on delta.
    b. Define delta = .1:.01:1
    c. Write a for loop to run the equations for every value of delta
        i. for i = 1:length(delta)
        delta1 = delta(i);
    d. Input T*, which changes with each delta
    e. Set tspan = to the vector of given temporal data points.
    f. Separate the output of the solution matrix into vectors containing the solutions of T*, VI, VN and set Vtotal = VI + VNI
        i. for j = 1:length(u)
        Tstar(j) = u(j,1);
        VI(j) = u(j,2);
        VNI(j) = u(j,3);
        Vtotal(j) = VI(j) + VNI(j);
        end
    g. Compute the sum of the squares (I suggest using (log(Vtotal(n)) – log(Vdata(n)))^2 because the numbers are so big).
    h. Save the value of the sum of the squares in a vector for each value of delta
    i. End the for loop

j. Use command [y I] = min( ) to locate the smallest entry in the sum of squares vector
k. Print out the value of delta that give the best fit
l. In the 'ODE' function, simply change delta to delta1
m. Use the newly found value of delta in the code you wrote in step #1 tp plot the computed solution and the given data on the same set of axes.