function [ fitnessy ] = fitness( chrome, INPUTS )

%FITNESS The function that computes the fitness of a given chromosome that

%          tries to sort itself.

%

%   THE RULES: To determine the fitness the chromosome that is passed in is

%                a vector of ones and zeros every set of three booleans

%                represents either a head or tail of an arrow that makes a

%                comparision within the INPUTS that try to sort the INPUTS.

%                The fitness is incremented if it can sort the INPUTS, but

%                only if it is not already sorted.

%Initiates the fitness to zero because it has not sorted the input.

fitnessy = 0;

%These are the heads and tails of the arrows that represent the compare and

%exchange pairs.

heads = zeros(1,length(chrome)/6);

tails = zeros(1,length(chrome)/6);

%Copies the chromosome so it can sort out how it should run, then gets the

%heads and the tails of the arrows.

tchrome = chrome;

for i = 1:length(chrome)/6

    heads(i) = nospaces(dec2bin(tchrome(1:3)));

    tails(i) = nospaces(dec2bin(tchrome(4:6)));

    tchrome = tchrome(7:end);

end

clear tchrome;

%Sets a count flag that determines whether or not the fitness should be

%counted if the input is not already sorted.

parfor i = 1:length(INPUTS)

    tmpIN = INPUTS(i,:);

    for j = 1:length(tails)

        %If the tail of the arrow is greater than the point then the two

        %need to be switched.

       if tmpIN(tails(j)) > tmpIN(heads(j))

           %A copy of the tail for future use.

           tmp = tmpIN(tails(j));

           %Changes the tail to the lower head.

           tmpIN(tails(j)) = tmpIN(heads(j));

           %Changes the head to the old tail.

           tmpIN(heads(j)) = tmp;

       end

    end

    %Increases the fitness if it has been sorted and if COUNT is true, mean

    if issorted(tmpIN)

        fitnessy = fitnessy + 1;

    end

end

end