function [ new\_Chrome ] = newchromes( old\_Chrome, fitnesses, gen, PROB\_MUT, PROB\_CRO, BEST\_IND, PROB\_MULT\_MUT, MAX\_MUT )

%NEWPOP Gets the new population of chromosomes from the most fit chromosome

%that has been found to have the greatest fitness.

%   The new population of chromosomes will be comprised of chromosomes that

%   mutants or crossovers from the most fit chromosome.

    %This is a flag that says whether or not to use the best individual

    %chromosme originally set to false, because we are not sure yet if we

    %would rather use the best individual.

    USE\_BEST = 0;

    %This where it is says whether a better individual has been found or

    %not and whether or not to use the already best individual as opposed

    %to the population that was found from the previous best individual.

    if max(fitnesses(gen,:))<max(fitnesses(:))

        USE\_BEST = 1;

    end

    %'unused' is unused should equal to gen for debugging purposes.

    %'keep\_indices'is the array of column indices for the chromosome that

    % should be kept based on if they are more fit than the previous

    % generation.

    [unused keep\_indices] = find(fitnesses(gen,:)>=(max(fitnesses(gen,:))));

    [unused max\_index] = find(fitnesses(gen,:)==max(fitnesses(gen,:)),1);

    %Size of the population.

    %size\_P(1) should be 1

    %size\_P(2) length of the chromosome

    %size\_P(3) number of the chromosomes

    size\_P = size(old\_Chrome(1,:,:));

    new\_Chrome = zeros(size\_P);

    %Len is the length of indices vector that we want to keep.

    len = length(keep\_indices);

    %This is a test to see if the number of keeper chromosomes is same as

    %the number of chromosomes then only use a third of them therefore

    %there is some variation in the next generation for improvements.

    if len == size\_P(3)

        len = ceil(len/3);

    end

    %This is how many mutations that we would like to happen when a

    %chromosome just so happens to mutate. Found that one seems to work the

    %best and is most efficient. But given a certain probability of more

    %than one mutation taken place and a max number of mutations more than

    %one might happen. But initially only one.

    mutations\_per\_chrome = 1;

    if rand()< PROB\_MULT\_MUT

        mutations\_per\_chrome = ceil(rand()\*MAX\_MUT);

    end

    %Builds the chromosomes by first getting all the chromosomes that we

    %want to keep from the old chromosome set then adds the rest by

    %mutating or crossing over with the keepers.

    for i = 1:size\_P(3)

        %This makes sure that at least all the keepers are in the new

        %chromosomes

        if ~USE\_BEST

            %This just keeps chromosomes from previous generations.

            if i <= len

                new\_Chrome(1,:,i) = old\_Chrome(1,:,keep\_indices(ceil(rand()\*len)));

            %Does mutations of keeper chromosomes.

            elseif rand() < PROB\_MUT

                new\_Chrome(1,:,i) = mutate(old\_Chrome(1,:,ceil(rand()\*len)), ...

                    mutations\_per\_chrome);

            %Does crossovers from the keeper chromosomes.

            elseif rand() < PROB\_CRO

                temp\_In = [keep\_indices(ceil(rand()\*len)),...

                    keep\_indices(ceil(rand()\*len))];

                new\_Chrome(1,:,i) = crossover(old\_Chrome(1,:,temp\_In(1)),...

                    old\_Chrome(1,:,temp\_In(2)));

            %If it does not mutate or crossover it duplicates the max

            %chromosome.

            else

                new\_Chrome(1,:,i) = old\_Chrome(1,:,max\_index);

            end

        %Use the best and make new chromosomes that are only identical or

        %mutations or crossovers of the best individual.

        else

            %Mutates the best individual chromosome

            if rand() < PROB\_MUT

                new\_Chrome(1,:,i) = mutate( BEST\_IND, mutations\_per\_chrome);

            %Does crossovers from the keeper chromosomes.

            elseif rand() < PROB\_CRO

                new\_Chrome(1,:,i) = crossover(BEST\_IND,BEST\_IND);

            %If it does not mutate or crossover it duplicates the best

            %chromosome.

            else

                new\_Chrome(1,:,i) = old\_Chrome(1,:,max\_index);

            end

        end

    end

end