```
function childrenPostMutation = GreenModMutation(children)
%   INTRODUCES RANDOM MUTATION INTO THE ARRAY OF CHROMOSOMES AFTER
%   SELECTION AND CROSSOVER. MUTATED GENES ARE RANDOMLY DISTRIBUTED OVER
%   THE CHROMOSOME.
%
%   [MxN DOUBLE] = GreenModMutation([MxN DOUBLE]
%
%   INPUTS:
%       children : ARRAY OF POPULATION CHROMOSOMES AFTER SELECTION AND
%       CROSSOVER SEQUENCES HAVE BEEN IMPLEMENTED
%
%   OUTPUTS:
%       childrenPostMutation : RESULT ARRAY OF CHROMOSOME VECTORS AFTER
%       RANDOM MUTATION
%
%   ENGINEERS: JAMES COLLINS & NAMKHA NORSANG
%   PROJECT: ME 6101 GREEN MODULAR DESIGN GROUP PROJECT
%   DATE: NOVEMBER 2017
%   LOCATION: GEORGIA INSTITUTE OF TECHNOLOGY. ALT, GA
%
%   NOTE: THIS FUNCTION IS BASED ON MATLAB'S mutationuniform.m FUNCTION

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    [PopRow,chromosomeLength] = size(children);
    % SET 5% MUTATION RATE
    mutationRate = 0.05;

    % PERFORM MUTATION EVALUATION OVER ALL CHILDREN
    for i=1:PopRow
        % RANDOMLY DETERMINE THE INDICES THAT COULD BE SELECTED FOR A
        % MUTATION
        mutationPoints = find(rand(1,chromosomeLength)...
            < mutationRate);
            % LOOP THROUGH THE SELECTED MUTATION POINTS TO DETERMINE NEW
            % VALUES
            for j=mutationPoints
                % DERIVE A RANDOM NEW ELEMENT VALUE BETWEEN 1-37
                % IE BETWEEN 1 AND 37 COMPONENTS IN THE DESIGN VECTOR
                mutValue = randi(chromosomeLength);
                % IF THE RANDOMLY DETERMINED MUTATION ELEMENT VALUE IS THE
                % SAME AS THE EXISTING ELEMENT, REDO UNTIL A DIFFERENT
                % VALUE IS REACHED
                while mutValue == children(i,j)
                    mutValue = randi(chromosomeLength);
                end
                % ASSIGN NEW VALUE
                children(i,j) = mutValue;
            end
    end
    % INITIALIZE OUTPUT FROM MODFIED INPUT ARRAY
    childrenPostMutation = children;
end
```

Published with MATLAB® R2017b