

```

function [chromosomeArr,binChromosomeArr,mk,m] = chromoSort(chromosome)
% CHROMOSOME OUTPUT FORMATTER FOR OUTPUT OF GENETIC ALGORITHM TO BE
% PROCESSED BY THE FITNESS FUNCTION F(X).
% THE FUNCTION IS USED IN THE FALL 2017 ME 6101 FINAL GROUP PROJECT ON
% GREEN MODULAR DESIGN OPTIMIZATION
%
% NOTE: TO OBTAIN THE FORMATTED CHROMOSOME USED IN F(X), USE chromoSort.m
%       INSTEAD!
%
%
% INPUT: [1xN DOUBLE] CHROMOSOME OUTPUT FROM GA
%         EACH INDICE IN THE VECTOR IS A GENE INDICATING THE
%         NUMBER OF MODULES TO BE GROUPED. THE INDEX OF THE
%         GENE POSITION INDICATES THE NUMBER OF COMPONENTS IN
%         THE MODULE
%
% OUTPUT: [MxN DOUBLE]  REFORMATTED ARRAY BREAKING THE NUMBER OF
%         MODULES TO BE GROUPED INTO DIFFERENT ROWS.
%         THE INDICES MARK THE POSITIONS WHERE THE MODULE
%         NUMBERS ARE PRESENT - IE THE NUMBER OF COMPONENTS
%         IN THE MODULES
%         [MxN DOUBLE]  REFORMATTED ARRAY SIMILAR TO chromosomeArr BUT
%         CONSISTING OF BINARY ELEMENTS
%         [1x1 DOUBLE]  THE MAXIMUM NUMBER OF DIFFERENT ELEMENTS IN THE
%         INPUT CHROMOSOME VECTOR
%         [1x1 DOUBLE]  THE MAXIMUM ELEMENT VALUE CONTAINED IN THE
%         INPUT CHROMOSOME VECTOR
%
% ENGINEERS: JAMES S COLLINS
%             BEN DUSSALT
%             NAMKHA NORSANG
%             NISHANTH KATHIRVEL
%
% PROJECT: ME 6101 GREEN MODULAR DESIGN GROUP PROJECT
% DATE: NOVEMBER 2017
% LOCATION: GEORGIA INSTITUTE OF TECHNOLOGY. ATL, GA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L=length(chromosome); % GET THE LENGTH OF THE CHROMOSOME INPUT IE THE
% TOTAL NUMBER OF COMPONENTS

% IDENTIFYING MAXIMUM MODULE NUMBER IN CHROMOSOME AND DECLARING THE
% OUTPUT LOGICAL ARRAY
m=max(chromosome);
binChromosomeArr=false(L,L);
chromosomeArr=zeros(L,L);
% FOR LOOP TO SEPARATE GENES INTO INDIVIDUAL ROWS MARKING DIFFERENT
% COLUMNS
for i=1:L
    % GENERATING LOGICAL MASK ROWS
    binChromosomeArr(i,:)=chromosome==i;
    % PORTION TO ASSIGN THE DOUBLE ARRAY THE SEPARATE VALUES
    rw=zeros(1,L);
    rw(binChromosomeArr(i,:))=i;
    chromosomeArr(i,:)=rw;
end

```

```

% CREATION OF LOGICAL MASK IDENTIFYING
% ROWS CONTAINING ALL FALSE INDICES

%deleteMask=~any(binChromosomeArr,2);

% USE THE LOGICAL MASK TO IDENTIFY AND ISOLATE ALL THE ROWS CONTAINING
% TRUE ELEMENTS IN THE COLUMNS OR DOUBLES ~= 0

%%binChromosomeArr(deleteMask,:)=[];

%chromosomeArr(deleteMask,:)=[];

% DETERMINE THE PARAMENTER m USED IN LATER CALCULATIONS AS THE NUMBER
% OF FILLED ROWS IN THE CHROMOSOME ARRAY
[mk,~]=size(chromosomeArr);
% CONVERT THE LOGICAL TO TYPE DOUBLE FOR FITNESS FUNCTION CALCULATION
binChromosomeArr=chromosomeArr~=0;
binChromosomeArr=double(binChromosomeArr);

```

end