Complement.m

```
function Mtx_comp = Complement(Mtx_in, nbrvar)
    % Step 1 check if matrix is empty
    if isempty(Mtx_in)
        Mtx_comp = ones(1, nbrvar)*3;

    % Step 2 check if there is one 'don't care' cube
    elseif any(all(Mtx_in == 3), 2)
        Mtx_comp = [];

    % Step 3 is the matrix contain only 1 cube, find the complement
    elseif size(Mtx_in,1) == 1
        % Get index where 1 need to be replaced by 2
        rowOne = find(Mtx_in == 1);
        colOne = rowOne;
        indexOne = sub2ind([nbrvar nbrvar],rowOne,colOne);

        % Get index where 2 need to be replaced by 1
        rowTwo = find(Mtx_in == 2);
        colTwo = rowTwo;
        indexTwo=sub2ind([nbrvar nbrvar],rowTwo,colTwo);

        % Create matrix filled with 3
        Mtx_comp = ones(nbrvar,nbrvar)*3;
        % Place 2 where 1 was before and place 1 where 2 was before
        Mtx_comp(indexOne)=2;
        Mtx_comp(indexTwo)=1;
        % Find and remove rows that contains only 3
        Mtx_comp(find(all(Mtx_comp == 3, 1)),:) = [];
    end
end
```

test.m

```
clc;
% Load Matrix
A = dlmread('etape1OK.cubes');
B = dlmread('etape2OK.cubes');
C = dlmread('etape3OK.cubes');

% Find Complement Functions
A_N = complement(A,5)
B_N = complement(B,4)
C_N = complement(C,6)
```