# CHEN3600 – Computer-Aided Chemical Engineering Spring 2012

# Chemical Engineering Department CQ2

**T.D. Placek Auburn University**

**CQ2 – Filling Vectors with Randomly Selected Values**

DO NOT SUBMIT YOUR SOLUTION!

Create a function file named email\_Random\_Fill. The following code should be placed in the file:

function placetd\_Random\_Fill()

% The Random\_Fill Function will select "n" values randomly between "a" and

% "b" without repeating a value. If it is not possible to fulfill the

% request, a vector of length "n" should be returned containing all zeroes.

disp(Random\_Fill(1,10,1,5));

disp(Random\_Fill(1,10,1,10));

disp(Random\_Fill(1,10,1,10)); %tests to see that the arrangement is unique

disp(Random\_Fill(100,200,5,5));

disp(Random\_Fill(10,20,0.5,15));

disp(Random\_Fill(10,20,0.5,50)); % tests to see that zeroes are returned

end

Write the necessary MATLAB subfunction to perform the action indicated in the description.

Hint: There are two approaches (at least) that can be employed. You may wish to try them both for practice.

Method (1): Generate a random number using a,b,s and see if it is not yet in your vector. If true, place it in the vector and continue looking for the next unique value until you satisfy the constraint “n”.

Method (2): Generate a vector which contains all possible values using a,b,s. Make a random selection between 1 and n and place the corresponding value from the vector of available values in your return vector. Replace the value in your vector of available values with the last value in the vector and reduce its length by one. Select from the remaining n-1 values, etc.

function x = Random\_Fill(a,b,s,n)

% Method: Select from remaining choices

z = [a:s:b];

nz = length(z);

% check for early return

if n>nz

x=zeros(1,n);

else

for k = 1:n

move = randi(nz); % chose which to move

x(k)=z(move); % copy to next x location

z(move)=[]; % close ranks

nz=nz-1; % ready for next move

end %for

end %if

end %function

function x = Random\_Fill2(a,b,s,n)

% Method: Guessing to find next choice

z = [a:s:b];

nz = length(z);

clear z;

% check for early return

if n>nz

x=zeros(1,n);

else

guess = myrand(a,b,s);

k=1;

x(k)=guess; % first must be unique

while length(x)~=n

guess = myrand(a,b,s);

if ~any(x==guess) % found a unique value... save it

k=k+1;

x(k)=guess;

end %if

end %while

end %if

end %function

function rd = myrand(a,b,s)

n=(b-a)/s+1; %number of states

m=randi(n)-1; %number of actual intervals (counting from zero)

rd = s\*m+a;

end