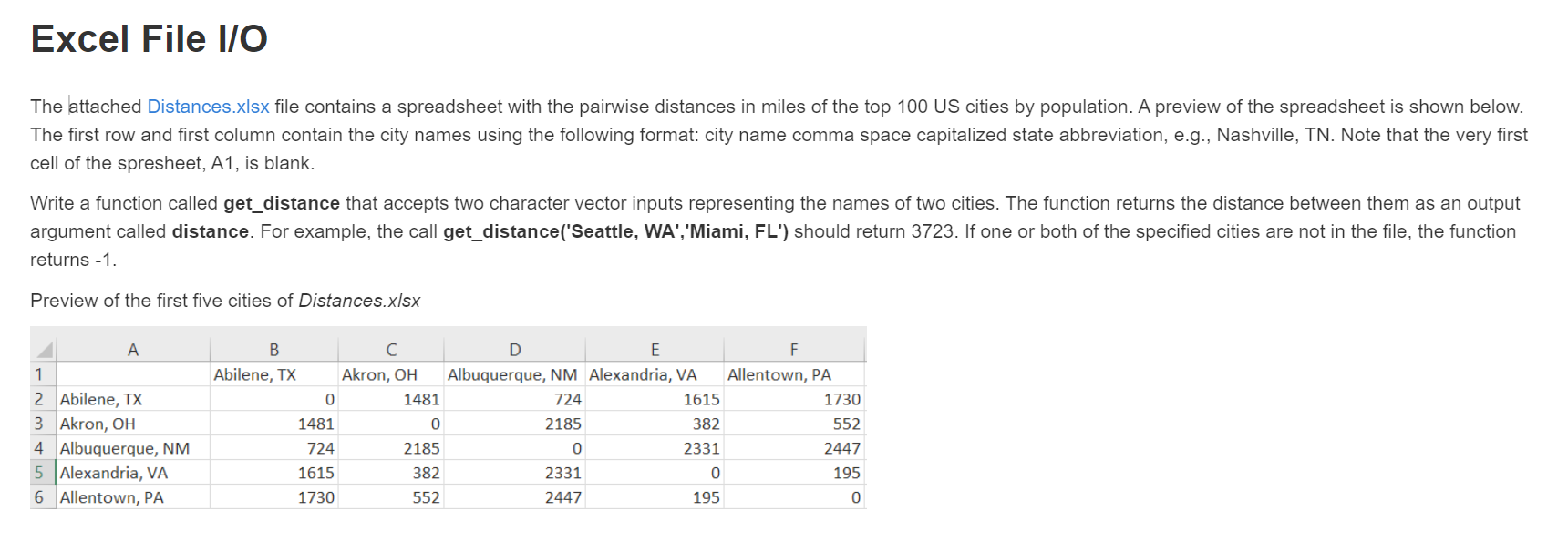
Coursera : Introduction to MATLAB programming

Week 9 : Lesson 8 : Files



Ans:

function distance = get\_distance(a,b)

found = true;

[~,text,raw] = xlsread('Distances.xlsx');

for i=2:size(raw,1) % this size function gives the total no of rows in it

for j=2:size(raw,2) % this size function gives the total no of columns in it

if strcmp(text{i,1}, a) && strcmp(text{1,j}, b) % compares the string given in the output

distance=raw{i,j};

found = false;

break

end

end

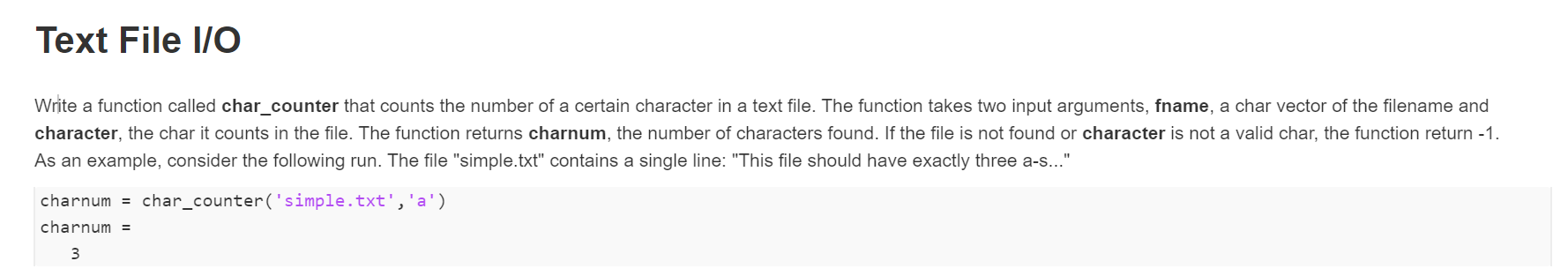
end

if found

distance = -1;

end

end



Ans:

function charnum = char\_counter(fname,char)

fid = fopen(fname,'rt');

if fid<0

charnum = -1;

return

end

if ischar(char) == 0 % if the input is not a char to be found

charnum = -1;

return

end

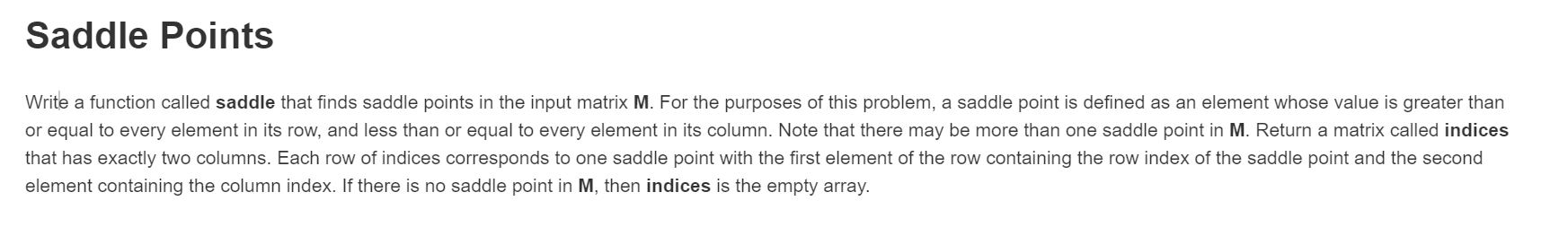
format = '%c';

A = fscanf(fid,format); % read all the char in the file and stores in the column vector A.

charnum = count(A,char); % count the no of times char appears in file

fclose(fid);

end



Ans :

%this may be helpful for the beginners

function indices=saddle(M)

indices=[];

n=1;

[row,col]=size(M);

max\_row=max(M,[],2)'; %finding the maximum element of each row and storing it in a vector

min\_col=min(M,[],1); %finding the minimum of each column and storing it in a vector

for r=1:row

for c=1:col

%as the saddle\_point\_element ,S>=every element in the row i.e. S=maximum of the elements of the row elements

%and S<=every element in the column i.e. S=minimum of the column elements

if (M(r,c)>=max\_row(r)) && (M(r,c)<=min\_col(c))

%instead of this line you may also use

% if(M(r,c)==max\_row(r)) && (M(r,c)==min\_col(c))

indices(n,1)=r;

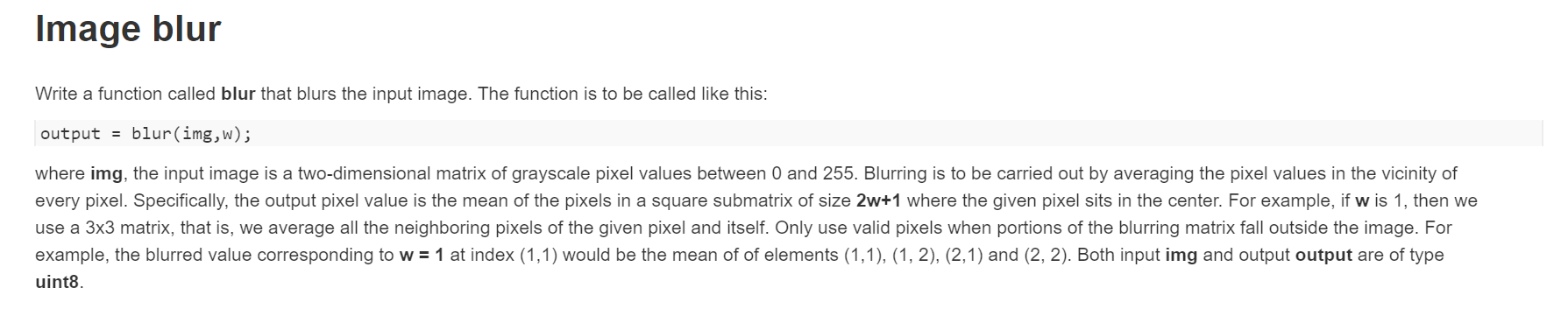
indices(n,2)=c;

n=n+1;

end

end

end



Ans:

function [output] = blur(A,w)

[row col] = size(A);

A=uint8(A);

B=nan(size(A) + (2\*w));

B(w+1:end-w,w+1:end-w)=A;

output = 0\*A;

for i=w+1:row+w

for j=w+1:col+w

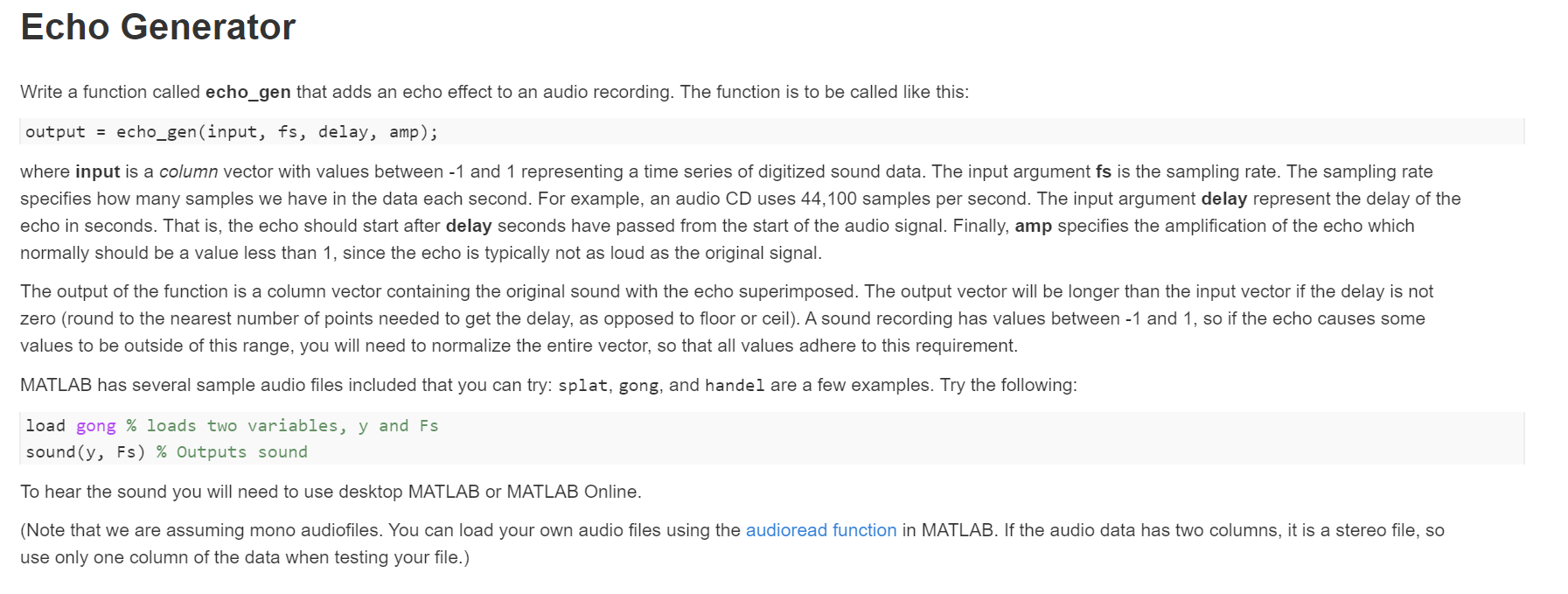
tmp=B(i-w:i+w,j-w:j+w);

output(i-w,j-w)=mean(tmp(~isnan(tmp)));

end

end

output=uint8(output);



Ans:

function output = echo\_gen(in,fs,delay,gain)

samples = round(fs\*delay) ;

ds = floor(samples);

signal = zeros(length(in)+ds,1);

signal(1:length(in))=in;

echo\_signal =zeros(length(in)+ds,1);

echo\_signal(ds+(1:length(in\*gain)))=in\*gain;

output= signal + echo\_signal;

p= max(abs(output));

if p>1

output=output ./ p;

else

output = output;

end

end