GM04: MATLAB Exercise 2 Solution

```
%%%%% By N. Ovenden, T. L. Ashbee and A. Whitfield

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q1: Loops
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A = zeros(5,7);
for ii = 1:5
      for jj=1:7
          A(ii,jj) = ii+jj;
      end
end
A

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q2: Conditional statements
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% note it makes no difference if 0 is regarded as
% positive or negative

x = [-15, -6, 1, 0, 8, -2, 5, 4, -10, 0.5, 6];
pp = 0;
nn = 0;

for ii = 1: length(x)
    if x(ii) > 0
        pp = pp + x(ii);
    else
        nn = nn + x(ii);
    end
end
```

```
pp
nn

% alternatively use logical indexing
% note 2 lines of code compared to 11 using a loop
P=sum(x(x>0))
N=sum(x(x<=0))

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q3: Conditional statements
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

-------------------------------------------------------
----------------- m-file CartToPolar.m ----------------

function [r, theta] = CartToPolar(x, y)

    r = sqrt(x^2 + y^2);

    if x >0, theta = atan(y/x);

    elseif x < 0 && y>= 0, theta = atan(y/x) + pi;

    elseif x < 0 && y < 0, theta = atan(y/x) - pi;

    elseif x == 0 && y > 0, theta = pi;

    elseif x == 0 && y < 0, theta = -pi;

    else error('undefined');

    end

end

-------------------------------------------------------
-------------------------------------------------------
```

```
% example call

[r, theta] = CartToPolar(1,0)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q4: Conditional statements
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A = [5 3 1 10 22 7; 2 3 6 1 13 5; 6 14 1 30 28 10]

% declare 'a'
% note can not pre-define size of array, as number of
% elements are unknown
a = [];

% find all numbers larger than 10
% loop through the entries of the matrix
for ii = 1:size(A,1)
    for jj = 1:size(A,2)
        if A(ii,jj)>10
            a = [a;A(ii,jj)];
        end
    end
end

a


% find all numbers that differ from the mean by more
% than 7
m = sum(sum(A))/numel(A);

% declare 'b'
b = [];

% loop through the entries of the matrix
for i = 1:size(A,1)
```

```
    for j = 1:size(A,2)
        if abs(A(i,j)-m)>7
            b = [b;A(i,j)];
        end
    end
end

b

% find all numbers between 10 and 20 or within 5 of
% the mean value

% declare 'c'
c = [];

% loop through the entries of the matrix
for i = 1:size(A,1)
    for j = 1:size(A,2)
        if (A(i,j)>10 && A(i,j)<20) | abs(A(i,j)-m)<=5
            c = [c;A(i,j)];
        end
    end
end

c

% alternative is to use logical indexing
% note the compactness of the code
c2 = A((A>10 & A<20)|abs(A-m)<=5)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q5: Conditional statements
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

-------------------------------------------------------
----------------- m-file sort_matrix.m ----------------

function [a,b,c,d,e] = sort_matrix(matrix)
```

```
% function to sort the elements of a matrix into
% separate
% arrays defined by the ranges 0<=x<=5 (a), 6<=x<=10 (b),
% 11<=x<=15 (c), 16<=x<=20 (d), x>20 (e).

a=[]; b=[]; c=[]; d=[]; e=[];

for i=1:size(matrix,1)
    for j=1:size(matrix,2)
        if matrix(i,j) <= 5 &  matrix(i,j) >= 0
            a = [a; matrix(i,j)];
        elseif matrix(i,j) <= 10 &  matrix(i,j) >= 6
            b = [b; matrix(i,j)];
        elseif matrix(i,j) <= 15 &  matrix(i,j) >= 11
            c = [c; matrix(i,j)];
        elseif matrix(i,j) <= 20 &  matrix(i,j) >= 16
            d = [d; matrix(i,j)];
        elseif  matrix(i,j) >= 20
            e = [e; matrix(i,j)];
        end
    end
end

-------------------------------------------------------
-------------------------------------------------------

% example call
[a,b,c,d,e] = sort_matrix(A)

--------------------------------------------------------
---------- m-file sort_matrix_switch_case.m ------------

function [a,b,c,d,e] = sort_matrix_switch_case(matrix)

% alternatively use switch case

a=[]; b=[]; c=[]; d=[]; e=[];
```

```
for i=1:size(matrix,1)
    for j=1:size(matrix,2)
        switch 1 % if condition true case = 1, if
                 % condition false case = 0
            case {matrix(i,j) <= 5 &  matrix(i,j) >= 0}
                a = [a; matrix(i,j)];
            case {matrix(i,j) <= 10 &  matrix(i,j) >= 6}
                b = [b; matrix(i,j)];
            case {matrix(i,j) <= 15 &  matrix(i,j) >= 11}
                c = [c; matrix(i,j)];
            case {matrix(i,j) <= 20 &  matrix(i,j) >= 16}
                d = [d; matrix(i,j)];
            case {matrix(i,j) >= 20}
                e = [e; matrix(i,j)];
        end
    end
end

-------------------------------------------------------
-------------------------------------------------------

% example call
[a,b,c,d,e] = sort_matrix_switch_case(A)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q6: Printing to file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% note '\n' in a string is interpreted as new line

% open a file to write to
fileid=fopen('Q6.txt','w');

% add header to file
fprintf(fileid,'Values of matrix A \n');

% loop over all elements of matrix
for ii=1:3
```

```
    for jj=1:6
    fprintf(fileid,...
        'Row: %3.0f \t Column: %3.0f \t Value: %3.0f \n',...
           ii,jj,A(ii,jj));
    end
end

% close the file
fclose(fileid);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q7: Logical indexing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

-------------------------------------------------------
-------------- m-file sort_vector.m -------------------

function [d_by_2,d_by_3,not_d]=sort_vector(intvec)
% function that sorts the entries of a vector of
% integers into numbers
% divisible by 2 and numbers divisible by 3 and others.
% Numbers divisible by both appear in both lists.

% divisible by 2
d_by_2 = intvec(mod(intvec,2)==0);

% divisible by 3
d_by_3 = intvec(mod(intvec,3)==0);

% others
not_d = intvec(mod(intvec,2)~=0 & mod(intvec,3)~=0);

end

-------------------------------------------------------
-------------------------------------------------------

% example call
```

```
[d_by_2,d_by_3,not_d]=sort_vector(A)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% Q8:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

-------------------------------------------------------
-------------- m-file sort_vector.m -------------------

function [z] = gauss_func(x,y,sigma)

z = exp(-0.5*(x.^2 + y.^2)*sigma^2);

end

-------------------------------------------------------
-------------------------------------------------------


x = -2:0.1:2;
y = x;

z = zeros(length(x), length(y));

for ii=1:length(x)
      for jj=1:length(y)
          z(ii,jj) = gauss_func(x(ii), y(jj), 0.1);
      end
end

[xx,yy]=meshgrid(x,y);
zz = gauss_func(xx, yy, 0.1);

% output to file twoDgauss.dat
% open file for writing
fileid = fopen(’twoDgauss.txt’,’w’);

% print the column headings
fprintf(fileid,’x \t y \t z \n’);
```

```
% print results
for i=1:length(x)
    for j=1:length(y)
        fprintf(fileid,'%.3f \t %.3f \t %.3f \n',xx(i,j),yy(i,j),zz(i,j));
    end
end

% close the file
fclose(fileid);
```