Max Kramer

Table of Contents

I affirm that I have adhered to the honor code on this assignment.

*Hello again, scientist! I'll write in italics, and problems for you will always be in **bold**. As a general rule, I expect you to do at least as much writing as I do. Code should be part of your solution, but I expect variables to be clear and explanation to involve complete sentences. Cite your sources; if you work with someone in the class on a problem, that's an extremely important source.*

Problem 1.2.

Googling MATLAB problems is often extremely helpful: the online documentation is amazing.

Problem 1.2a.

***Read the following example.** Let C be a 10x10 matrix of all zeros. Explain what C is.*

I worked with my friends Rebecca, Ryan, and Sean from Maryland on this entire problem set.

```
C = zeros(10) % Sean told us in Discord to Google "MATLAB zeros", and
 that helped me a lot.
% It makes sense to me and I tried my best to explain the result in my
 own words.*
```

```
C =

     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

There's not much to say here; this is in fact a 10x10 matrix of all zeros. Why did you ask me to explain this??^

** You don't have to tell me every time you or your friends Google something, but these problems will get harder and will eventually require serious and significant collaboration. When you find the answer to a big problem on a website or by going to an office hour or because someone posts a hint on Slack or by asking your CS major roommate, you need to tell me.*

^ Because this is an example to show you how to format explanations, that's why. You don't have to explain the output of each line of your code if the result is "obvious," but be warned that you and I might have different definitions of the word "obvious."

Example ends here. Have fun! Don't work alone.

Problem 1.2b.

Let D be a 9x4 matrix of all ones.

```
D = ones(9,4)

% The ones(r,c) command creates a matrix of dimensions (r,c) of all
 ones.
```

```
D =

     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

Let E be the 7x7 identity matrix.

```
E = eye(7)

% The eye(n) command creates an n x n identity matrix.
```

```
E =

     1     0     0     0     0     0     0
     0     1     0     0     0     0     0
     0     0     1     0     0     0     0
     0     0     0     1     0     0     0
     0     0     0     0     1     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     1
```

Let F be a 9x7 matrix which has 2s above and below the main diagonal and -1s on the main diagonal. *(Try to do this without typing 63 numbers.)*

```
F = ones(9,7) * 2;
f1 = triu(F,1);
f2 = tril(F,-1);
f3 = eye(9,7) * -1;
F = f1 + f2 + f3

% To create the matrix F, we begin with a matrix of 2s with 9 x 7
% dimensions. This was done by multiplying a call to ones(r,c) by 2.
 Then,
% the triu (upper triangle) and tril (lower triangle) submatricies
 were
% created. Finally, a call to eye(r,c) and multiplying by -1 gave us
 the
% diagonal we needed. After that, simply adding all three submatricies
% together produced the desired matrix F.


F =

    -1     2     2     2     2     2     2
     2    -1     2     2     2     2     2
     2     2    -1     2     2     2     2
     2     2     2    -1     2     2     2
     2     2     2     2    -1     2     2
     2     2     2     2     2    -1     2
     2     2     2     2     2     2    -1
     2     2     2     2     2     2     2
     2     2     2     2     2     2     2
```

Let G be a 4x5 matrix of integers drawn randomly from 0:9. *If you don't know what 0:9 means, uncomment the following line. Now you do!*

```
G = randi(10,4,5) - 1

% The randi(n,r,c) command creates an r x c matrix of uniformly drawn
% random numbers in the range [1,n]. The [1:10] range is corrected to
 [0:9]
% by subtracting 1 from all numbers in the matrix.


G =

     4     6     0     4     5
     0     6     3     8     3
     5     6     5     7     1
     4     0     6     9     6
```

Problem 1.2c.

One thing that makes coding in MATLAB unique is that many functions are heavily ***overloaded****, meaning that they do different things depending on how you call them. Once you've found G,* ***uncomment the following code and then explain what's going on.***

```
v = diag(G)
diag(v)

% The first line `v = diag(G)` returns a column vector of the diagonal
% elements of the matrix G. The second line `diag(v)` creates a new
 matrix
% where the elements of v are placed on the main diagonal.
```

```
v =

     4
     6
     5
     9


ans =

     4     0     0     0
     0     6     0     0
     0     0     5     0
     0     0     0     9
```

Problem 1.2d.

Okay, now let's do something with a specific matrix.

```
H = [1 2 3 4; 5 6 7 8; 9 10 11 12]
```

```
H =

     1     2     3     4
     5     6     7     8
     9    10    11    12
```

Let b be the last column of H. *(Please don't just type [4;8;12].)*

```
b = H(:,end)

% To get all rows and the last column of a matrix, we call the matrix
 and
% supply a : for all rows and then use `end` to get the last column.
```

```
b =

     4
     8
    12
```

Let J be H stacked vertically on top of H.

```
J = vertcat(H,H)

% The vertcat(n,m) command stacks n on top of m if n and m have
 compatible
% sizes.
```

```
J =

     1     2     3     4
     5     6     7     8
     9    10    11    12
     1     2     3     4
     5     6     7     8
     9    10    11    12
```

Published with MATLAB® R2019b