

Week One Answers

Feel free to copy the code provided into your own MATLAB scripts to check your work! Please remember that although these are answers, there are multiple ways of doing the problem(s). We encourage you to develop your own method of writing and organizing your code! - [Lectures](#) - [Practice Problems](#)

Question 1

```
```MATLAB %Basic Matrix Problems (Week 1)

%Defining the matrix mat_1 = [1 3 5; 2 6 8; 4 7 9];

%1 %1a: ans1a = mat_1 .* 3

%1b: ans1b = mat_1 + 2

%1c: ans1c = mat_1(1,:);

%1d: ans1d = mat_1(:,3)

%1e: ans1e = mat_1(2,3)/2 -5

%1f: ans1f = mat_1 ans1f(3,2) = ans1e ```
```

**Notes:** \ Observe how we define the matrix. A comma is not necessary when defining a matrix, as well as the use of semi-colons to create the next row.\ Another thing to note is how we call select elements and copy matrices. To call a specific element in a matrix, we must call it by it's **row** then **column**.\ The colon ( : ) is used in matrices to call **ALL** elements in a row or column. [MathWorks: Array Indexing](#)\ Finally, just like Python, if we would like to create a multiple of an element or matrix, we can make a new label for the matrix and copy the matrix and make modifications to the new one without affect the old one.

### Question 2

```
``` MATLAB %Basic Matrix Problems 2 (Week 1) mat_2 = [5 1 7; 2 6 3; 4 8 9];

%2 %2a: ans2a = mat_1 * mat_2

%2b: ans2b = mat_1 .* mat_2

%2c: ans2c1 = mat_1 + mat_2 ans2c2 = mat_1 - mat_2

%2d: ans2d = mat_1(1,2) + mat_2(2,3)

%2e: ans2e = mat_1 * mat_2(:,2) ```
```

Notes:\ Observe the Difference between the notation for matrix multiplication (*) and element-wise multiplication (.*)\ As a general rule, when attempting to do matrix multiplication, the number of **columns** of the first matrix must match the same number **rows** of the second. And order matters!

EX: \ 2 x 3 * 3 x 1 Works!\ 3 x 1 * 2 x 3 Does not work!

One Final thing is that if you would like to add or subtract matrices together, they must have the exact same dimensions!

Question 3 (For Loop)

```
``` matlab %For Loop Exercise (Week 1)

test = randi([0,1],1,10); %Customize to Test! counter = 0;

for i = 1: length(test) if test(i) == 1 counter = counter + 1; end

if i == length(test)
 disp(counter)
end

end ```
```

**Notes:** \ A good habit when writing any conditionals or functions in MATLAB is to immediately go to the next line and type **end**. This way, when you are typing out your code, you know what commands are in which conditional.\ You can change the 10 in the **randi** command to any number to test to see if your for loop works or not!\ [MathWorks: For Loops](#)

### Week 1 Function

```
``` matlab function [ans1a, ans1c, ans1d, ans2a, ans2b, ans2c1, ans2c2, ans2e] = ansW1fun(mat_1,mat_2) ans1a = mat_1 .* 3;

ans1c = mat_1(1,:);
```

```
ans1d = mat_1(:,3);
```

```
ans2a = mat_1 * mat_2;
```

```
ans2b = mat_1 .* mat_2;
```

```
ans2c1 = mat_1 + mat_2; ans2c2 = mat_1 - mat_2;
```

```
ans2e = mat_1 * mat_2(:,2); end ``
```

Notes: \ In this function, I just copied the answers from the respective parts and placed them in. \ Remember that ***Functions must be in their own file (.m file)***. In order to call the function, the file must be in your current folder. \ In this case, since there are multiple returned values. In order to call this function to return all of the answers, it would be called the same exact way it is defined.

```
matlab [ans1a, ans1c, ans1d, ans2a, ans2b, ans2c1, ans2c2, ans2e] = ansW1fun(mat_1,mat_2)
```

You can change the variable names of the answers when you call it here. \ [MathWorks: function](#)