Introduction to Programming with MATLAB

Frequently Asked Questions about the Homework

When should I read this document?

You should read this document before attempting to solve the programming assignments and if and when you run into a problem doing so.

Should I watch Lecture 3.8 Problem Solving before working on the homework?

Yes! You will save a lot of time for yourself if you watch the video and carefully follow the advice therein.

I have an old MATLAB version already installed. Can I use it?

Any version 2012a and newer will work fine. The grader will not work correctly with older versions though. But you can have multiple versions of MATLAB installed simultaneously on your computer. So, download the latest one and use it for this course!

I downloaded the grader (hw3.p), but when I double click on it, I get an error message.

You need to follow the instruction to the word. Save the grader into an empty folder. Make sure that its name is unchanged, for example, hw3.p as opposed to hw3(1).p in case you downloaded it multiple times. Start MATLAB and set its current folder to the folder where you have the grader. In MATLAB, in the Current Folder window, you should be able to see the file now. Do not click or double click it. Type hw3 in the command window and the grader will start.

Can I post my code on the forum?

Absolutely not! Everybody is here to learn and you can only do it if you solve the assignments yourself. Using somebody else's solution may get you a certificate, but it won't make you a programmer.

How many times can I submit my alphanumeric code?

You can submit it up to 1000 times. And only the best one will count. So, do not wait until the last minute before the deadline. If you have already solved a few problems, submit the code. If you solve more before the deadline, submit the new code and you will get credit. But the deadline is hard. So, if you miss it, you will get no credit at all. Also note that we cannot grant deadline extensions for any reason. With tens of thousands of students, it is simply not feasible.

I solved the problem, my function is correct, yet the grader says it is incorrect.

First of all, when you think you solved the problem, test your function with different inputs directly in MATLAB. That is, call your function multiple times with various input arguments. Let's say, I want to test the built-in sum function to make sure that it works correctly. I would try it with a scalar, a column vector, a row vector, a matrix and an empty array. Something like this:

```
>> sum(3)
ans =
     3
>> sum(1:3)
ans =
     6
>> sum([1 2 3]')
ans =
     6
>> sum([1:3;4:6;7:9])
ans =
    12    15    18
>> sum([])
ans =
     0
```

When it seems to work for various inputs, only then should you try it with the grader.

I did all that, but the grader says that my function failed with an input of [1 2; 3 4]. What should I do?

Try your function with the exact same input directly in MATLAB and see what happens. You will either get an error message from MATLAB, such as, Subscript indices must either be real positive integers or logicals or you will get a result that is not what the assignment asks for. Think hard to figure out why your result is incorrect and when you get it, work hard to fix the problem in your function. Once you think you have done it, test your function again in MATLAB before running the grader again.

I did all that, but I cannot figure out what the problem is. What should I do?

It might be time to ask a question on the Discussion Forum. But you need to do it in the right subforum and it needs to be as informative as possible. There is a subforum for each week's homework and under it, there is a subforum for each individual problem. Find the right one! Then read the discussion because chances are that other people have already run into the same problem and there is an answer to your question already posted. If not, post a new question. In it, copy and paste what the feedback from the grader was and what MATLAB prints out when you run your function with the same input. Something like this:

The grader reports that my function fails. See below:

```
Problem 12 (add_it_all):
  Feedback: Your function performed correctly for argument(s) 1, 1
  Feedback: Your function did not return the correct answer(s) for
  argument(s) 2, 3
      Your solution is _not_ correct.
```

But this is what I get when I run it directly:

```
>> xy = add_it_all(2, 3)
xy =
          4
>> whos xy
  Name      Size            Bytes  Class     Attributes
  xy        1x1                 8  double
```

Such a post will provide enough information to the reader to help you out. They cannot see your code, so just saying that your solution is correct without any useful information would not be very helpful.

The problem asks for an input argument of a vector of three elements. I have that and my function computes the correct result, yet the grader complains. What is going on?

Many MATLAB novices have trouble with such assignments initially. Your function will have a single input argument, let's call it inp. Your function can assume that it has three elements exactly. If you read the assignment, you will see that it says that you can assume that the inputs are correct. When you test your function, that is, when you call it from the Command Window, that is when you need to make sure that the input has three elements. Here is how your function will look:

```
function result = multiply_elements(inp)
   result = inp(1) * inp(2) * inp(3);
```

The function gets inp with its elements and it uses these elements. It does not set inp! It is a bad idea to set the values of the input argument inside your function. That is your input, use its value, but do not change it! Nor does the function check that inp has three elements because we said it was not necessary for these initial assignments. And here is how you would check your function from the Command Window:

```
>> multiply_elements([1 2 3])
ans =
     6
>> multiply_elements([0 1 1])
ans =
     0
>> multiply_elements([2 2 2])
ans =
     8
```

You need to make sure that you call the function from the Command Window with a single input, a vector of exactly three elements (in this example).

My function works correctly, but the grader complains. What to do?

Here is another typical mistake:

```
function multiply_elements(inp)
   inp(1) * inp(2) * inp(3)

>> multiply_elements([1 2 3])
ans =
     6
```

It works, right? No, it does not! Printing the result by omitting the semicolon inside the function is not the same as returning the result. Your function has no output, so if we tried to assign the result to a variable, we would get an error:

```
>> x = multiply_elements([1 2 3])
Error using multiply_elements
Too many output arguments.
```

That is why the grader complains. See the correct solution under the previous question.

My function has two outputs, but MATLAB swallows the second. How can I get it?

If you do not save the outputs of a function to variables, MATLAB will only save the first one into the variable called ans. If you want more than one output, you need to save them to variables yourself. Consider this illustration using the max built-in function that finds the maximum value in a vector (or in each column of a matrix) and it also returns its index as a second output argument.

```
>> max(0:2:10)
ans =
    10
>> [mx mxi] = max(0:2:10)
mx =
    10
mxi =
     6
```

You need to use the same syntax when calling you own functions.

Do I need to use loops to solve Problem X in Homework 3?

No. Each homework tests your knowledge of that given Lesson's material. Especially for Assignments 3 and 4, there is an easy solution for each problem without loops or if-statements.