

# Practical 2 - MATLAB Programming Practice

- Due 5 Aug by 23:59
- Points 16
- Submitting an external tool

This practical exercise is to be prepared for in your own time and in your computer exercise class.

These questions are intended to be completed **individually** but you are allowed to consult other resources to answer generic questions that help you complete small parts of each question.

Please consult the **rubric** of this practical assignment for information on the marking scheme.

For this exercise you must:

1. Complete the exercises below. Do what you can prior to class and come into class to have questions answered.
2. Prior to the due-date submit your answers (MATLAB scripts and text files) via this assignment link in Gradescope.
3. You can choose to have your tutor mark you in person (recommended for feedback) in your practical or workshop session

## Question 1.

For this problem you have to create a program that calculates a person's Body Mass Index (BMI). The formula for BMI is:

$$\text{BMI} = \text{weight} / \text{height}^2$$

where *weight* is a person's weight in kilograms and *height* is a person's height in metres.

### Question 1a.

For this question you should, in a text file called, **BMI.txt** a table containing three test cases. As a reference example, two possible test cases for this are:

input: weight.	height.	expected output: BMI
70.0	1.55	29.14
65.0	1.77	20.75

Remember to write your test cases in the format above into your BMI.txt file and your three tests must be different from the examples above.

### Question 1b.

For this question you should write a MATLAB script, called, **BMIcalc.m**, that, when run, prompts the user for a weight then a height and prints out a formatted message that displays the BMI corresponding to these values to two decimal places after the decimal point. In your answer you will find, MATLAB's built-in input function useful for **input** and the **fprintf** function useful for output (see workshop 2 and this week's online lecture about how to program for information about the fprintf function).

As an example of how your script should behave a sample run might be:

```
Enter your weight in kgs: 70
Enter your height in m: 1.55
Your BMI is: 29.14
```

Can you think of input that might cause problems for your program?

## Question 2.

For this problem, you have to create a program that uses a loop to read in 5 numbers, which can be either positive or negative, and calculates the sum of the absolute values of the five numbers. Note, in answering the parts below you will find MATLAB's built-in **abs()** function useful. The abs() function takes a number and produces the positive version of that number. So, for example, the MATLAB code:

```
a=abs(-3); % will make positive
disp(a);
b=abs(5); % will stay positive
disp(b);
c=abs(-7); % will make positive
disp(c);
```

will output:

```
3
5
7
```

### Question 2a:

For this question, you should write tests for a program that calculates the sum of the absolute values of 5 numbers. In a text file called **sumAbs.txt** you must write three new test cases. As a reference example, two possible test cases for this are:

```
input: num1 num2 num3 num4 num5 expected output: absSum
-1.0 -2.0 3.0 -4.0 5.0 15.0
3.0 -7.0 -2.0 3.0 5.0 20.0
```

Remember to write your test cases in the format above into your sumAbs.txt file and create 3 new cases (not including the examples above).

### Question 2b.

For this question, you should write the MATLAB script **sumAbs.m** that uses a loop to read in 5 numbers, which can be either positive or negative, and calculates the sum of the absolute values of the five numbers. After calculating the sum, the script should print the sum with **one decimal place**. As an example of how the script might behave, a sample run might be:

```
enter a num: 3
enter a num: -7
enter a num: -2
enter a num: 3
enter a num: 5
the abs sum is: 20.0
```

Check your program works for your test cases as well as the example test cases.

## Question 3.

For this problem, you have to write a program that generates and adds together five random numbers. To generate a random number in MATLAB you can use a built-in function called **rand()**. The rand() function generates a random number between 0.0 and 1.0 for example. The MATLAB code:

```
myRand = rand();
disp(myRand);
```

will display a random number between 0 and 1.0 such as: 0.8147. Each time you run the code above it will (probably) display a different number.

But what if we want a random value that isn't between 0 and 1.0? For example, we might want to represent rolling a die with a random value between 1 and 6, inclusive. We might think of using maths to calculate a value in this range. For example, multiplying the rand value by 5 and adding 1 then rounding the result:

```
myThrow = round(rand()*5.0)+1; % the plus 1 stops the result being in the range 0..5
disp(myThrow);
```

But such mathematical conversions do not generate all numbers from 1 to 6 with even probability.

Try the code below to see if rand() can really simulate a die roll

```
counter = zeros(6, 1);

for i = 1:60000
    number = round(rand() * 5) + 1;

    if number == 1
        counter(1) = counter(1) + 1;
    elseif number == 2
        counter(2) = counter(2) + 1;
    elseif number == 3
        counter(3) = counter(3) + 1;
    else if number == 4
```

```
        counter(4) = counter(4) + 1;

    else if number == 5
        counter(5) = counter(5) + 1;

    else
        counter(6) = counter(6) + 1;
    end

end

disp(counter)
```

MATLAB has another built in function, `randi()`, which can generate a random *integer* directly. It can take two arguments, first sets the max integer it can generate, second is the number of random integers you need. For more details use `help randi` on your command window, or go to MATLAB documentation page.

```
myThrow = randi(6, 1);
disp(myThrow);
```

Likewise, if we wanted to generate a number that is either 1 or 2 (as we might do if we are trying to simulate the flip of a coin) then we could write:

```
myThrow = randi(2, 1);
disp(myThrow);
```

In this question you will write a program to sum 5 random numbers between 1 and 10 inclusive.

### Question 3a.

Because our program will generate a sum of random values we can't write tests in the normal fashion, this is because the program will give a different result every time it runs. However, you can still describe in words what sort of values you think the program will produce. For example, you might want to name the **maximum** value it could produce is and the **minimum** value, and perhaps the **average** value. Write your description in a text file called **sumRands.txt**. Note that, in this case, we are not looking for a detailed analysis or proof, just a quick description of your expectations (hint: you should be able to arrive at a close estimate of maximum and minimum and a reasonably close guess of average).

### Question 3b.

Write a MATLAB script called **sumRands.m** that uses a loop to generate 5 random integer values between 1 and 10 and sum them together and then displays their sum as an integer. As an example of this script's output 3 runs of `sumRands.m` might look like this:

```
>> sumRands
32

>> sumRands
40

>> sumRands
38
```

Because this function has random numbers, the results above will vary.

## Question 4.

For this problem, you have to write a program that reads in the heights of 5 students from the user and stores them in a MATLAB vector, and then prints out the values stored in the vector, with each value separated by a blank line. The test cases for this question are simple - the five inputs and five outputs are the same. A sample run for the program might be:

```
>> heightsVector
enter height 149
enter height 188
enter height 173
enter height 162
enter height 175
149

188

173

162

175
```

Recall from the vectors video for this week that we assign and read from vectors by indexing their elements with numbers. So for example we would assign to the first element of a vector heights with the code:

```
heights(1)=173;
```

and we would display the first element of vector h with:

```
disp(heights(1));
```

to match expected format the MATLAB `disp()` function should be used to print each array element. Write your solution in a MATLAB script called **heightsVector.m** and your test cases in **heightsVector.txt**

## End of Questions.

## Assesment

Marks are awarded for

- **functionality** (the code runs and produces expected results) - 8 marks
  - Note: 2 of these marks (for the code with random values) will be assigned manually by tutor on the Testing rubric in Gradescope
- **style** 4 marks (the code adheres to [MATLAB Style Guidelines](https://myuni.adelaide.edu.au/courses/95033/files/15473409?wrap=1) (<https://myuni.adelaide.edu.au/courses/95033/files/15473409?wrap=1>)\_ (<https://myuni.adelaide.edu.au/courses/95033/files/15473409/download>)\_ for this practical we will be assessing adherence to:

- **Naming Conventions:** for variables (all sections **other than** Structures, Functions.
- **Statements:** sections Variables and Constants and Loops, General: sections *before* "Use the natural, straightforward form for logical expressions".
- **Layout, Comments and Documentation :** all aspects other than those that mention *functions*
- **testing** for this practical we are looking for the presence of unique tests cases (3 per question) - 4 marks

This tool needs to be loaded in a new browser window

Load Practical 2 - MATLAB Programming Practice in a new window