MATLAB lesson 5: Programming
Exercise sheet

Dr. Gerard Capes*

1 Logical operators

Based on the lesson

1. Run the tests `a<b`, `a>=b`, `a==b`, and `a~=b` for the following values of `a` and `b`:

   (a) a=1, b=1

   (b) a=1, b=2

   (c) a=1, b=inf

   Make sure you understand the result.

2. For `a=1`, `b=1`, `c=0`, test the following and make sure you understand the results:

   (a) `(b | c) & (a < 2)`

   (b) `(a < b) | (b > c)`

   (c) `(a & b) & c`

3. Generate a 500 by 1 array (i.e. 500 rows, 1 column) of random values and test if any values are greater than 0.5 (hint: `rand`, `any`).

Also requires some use of MATLAB's help

4. Use the MATLAB help to find out how to write a run a script (hint: `doc Programming Scripts and Functions`).

5. Write a script that generates a 500 by 1 array (i.e. 500 rows, 1 column) of random values (use `rand`) and does the following:

   (a) Tests if any values are greater than 0.5, 0.9, 0.99

   (b) Finds the indices where values are greater than 0.99

*Questions and feedback can be directed to gerard.capes@manchester.ac.uk

(c) Finds if all values are greater than 0.5, 0.1 and 0.01

(d) Sets the values which are greater than 0.99 equal to 1

6. Repeat question 5 for a 10 by 10 matrix of random values For each test use the `disp` function to output sensible text.

7. Write a script which creates two 5*5 matrices, `r1` and `r2`, each filled with random numbers.

   (a) Test which of the values in `r1` are greater than their counterparts in `r2`.

   (b) Test which the values in `r1` are greater than 0.5, 0.9, 0.99. Your tests should return a 5*5 matrix of logical values.

2 Flow control

Based on the lesson

1. Create a variable `x` and assign it a value.

   (a) Write a script that uses an `if` statement to test whether the variable `x` has a value in the range $1 < x < 2$. The script should output a suitable message if it is (use `disp`).

   (b) Modify your script using an `elseif` to also test if `x` is less than or equal to 1. Your script should output a suitable message if so.

   (c) Add a further test which outputs suitable text when neither of the above conditions are met.

   (d) Change the value of `x` and re-run your script to test whether it works correctly for each condition.

2. (a) Write a script that uses an if statement to test whether the class of a variable `x` is a double (use the `isa` function). The script should output a suitable message if it is (use `disp`).

   (b) Modify your script using an `elseif` to also test for `char` and `logical` classes. Your script should output suitable message if `x` is any of these classes.

   (c) Modify your script to output the message "Unknown class" when the class of `A` is none of the above (e.g. `single`, `int8`).

   (d) Test your script by changing the class of `x` and re-running the script.

3. Write a new script that answers question 2.2 using the `switch` construct instead of `if`.

4. Consider why the solution to question 2.1 does not lend itself to being rewritten using a `switch` statement. When would you use a `switch` statement instead of a series of `if`, `elseif`, `else` statements?

5. (a) In a new script write a `for` loop that counts from 1 to 10. The script should output the value of the loop counter (to the screen) for each loop iteration.
   (b) Add a nested loop within the first loop that counts from 10 to 1.
   (c) Modify your code so both loops terminate when the loop counters are equal.

6. Write a while loop that repeatedly multiplies a variable `B` by 10 until `B` equals `inf`. If `B` is initially 1, how many loop iterations are required before the loop terminates?

Requires some use of MATLAB's help

7. Consider when you would use a while loop instead of a for loop (and vice-versa). Read the MATLAB help documentation if you're unsure (`doc for` and `doc while`)

8. Write a `for` loop which loops from 1 to 100 in steps of 1.
   (a) If the loop counter is divisible by both 5 and 7 (i.e. the answer is an integer), output suitable text to the screen. (hint: use the `mod` function and `fprintf`)
   (b) Count how many times the above condition is met and output the result

9. Write a while loop which calculates the first 10 numbers which are divisible by 3, 4 and 5. From within your loop, output sensible text to the screen for each number which meets all these criteria.

10. (a) Write a script which requests the user to input a number, then uses a loop to test whether this is a prime number. You should output some text to indicate the result, and if the number is not prime, indicate by which number is it divisible. (hints: `fprintf`, `input`)
    (b) Make your code more robust by checking that the user input is
        i. an integer
        ii. greater than 1

        and halting execution with a helpful error message if the input is invalid (hint: `assert` that these conditions are true)