

MATLAB CODING EXERCISES

FOR LOOP *for*

Programme 1:

- **Task:** Write a programme to display values increasing from 0 to 10 with a step of 1.

Programme 2:

- **Task:** Write a programme to display values increasing from 0 to 10 with a step of 2.

Programme 3:

- **Task:** Write a programme to display values decreasing from 10 to 0 with a step of 1.

Programme 4:

- **Task:** Write a programme that repeats the following instructions 3 times:
 - Ask the user of the programme to enter a number
 - Square the entered number
 - Display the result

Programme 5:

- **Task:** Write a programme to calculate a number factorial without using embedded Matlab function factorial.

(Example: $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$)

CONDITIONAL STATEMENT *if - else*

Programme 6:

- **Task:** Write a script to allocate a random value between 0 and 1 into variable R using the function *rand*.

If,

$$0.1 < R < 0.27$$

Or if,

$$0.72 < R \leq 0.9$$

The string '**BINGO**' should be displayed in the command window.

If value R does not belong to any of two ranges aforementioned than the string '**TRY AGAIN**' should be displayed in the command window instead.

Programme 7:

- **Task:** Write a programme that will keep returning the square root of user input values. The programme should be implemented using an infinite for loop that will be broken when the user does not need to enter another value.

Details of the code,

- Infinite for loop
- Ask user to input a numerical value
- Return the square root of inputted value
- Ask the user to input a string letter
- If the letter is 'n' or 'N' the infinite for loop will keep going and the user will be asked to input a new value.
- Else the code will implement the function *break* to stop the infinite for loop and exit the programme.

PLOTTING OF FUNCTIONS <i>plot</i>
--

Programme 8:

- **Task:** Write a script to plot the graph of following function s ,
- $$s(x) = 5 \times (x - 3)^2 + 10$$

And following function c ,

$$c(x) = (x + 2)^3$$

In the same figure.

Also,

- Your code should control where data appears in the axes by setting the x-axis and y-axis limits to $[-6, 8]$ and $[-30, 50]$ respectively.
- The square function should be plotted in green and the cubic function in blue.
- The x and y-axis should appear on the graph in red.
- Set the x-axis tick values, which are the locations along the x-axis where the tick marks appears, at increments of 1, starting from -6 and ending at 8.
- Display tick marks along the y-axis at increments of 5, starting from -30 and ending at 50.
- Determine the x and y coordinates at which the curves intersect.
- Return the coordinates of the intersection point in the command window.

BASIC PROBABILITY APPLICATION

Programme 9:

- **Task:** Write a code that achieves the following tasks,
 - a) Generates a random sequence of 50 flips (0 and 1) using the function `randi`.
 - b) Your code should return the longest **run** of the sequence where a **run** is a series of all 1's or 0's.
 - c) Implement a trial of 10,000 sequences. Estimate the average length of the longest run.
 - d) Estimate the probability of a run of 8 or more in a sequence over the trial of 10,000 sequences.